(2)

IDA PAPER P-2154

MANAGING ENGINEERING DESIGN INFORMATION

R. E. Fulton, *Georgia Institute of Technology*
Chou Pin-Yeh, *Georgia Institute of Technology*
Karen J. Richter, *IDA*

October 1989

Prepared for
Office of the Under Secretary of Defense for Acquisition
(Research and Advanced Technology)

Supported by
Air Force Human Resources Laboratory
Wright-Patterson AFB, Ohio

IDA

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under contract MDA 903 89 C 0003 for the Department of Defense. The publication of this IDA Paper does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

This Paper has been reviewed by IDA to assure that it meets the high standards of thoroughness, objectivity, and appropriate analytical methodology and that the results, conclusions and recommendations are properly supported by the material presented.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***

2. REPORT DATE
October 1989

3. REPORT TYPE AND DATES COVERED
Final 1/88 to 9/88

4. TITLE AND SUBTITLE
Managing Engineering Design Information

5. FUNDING NUMBERS
C- MDA 903 89 C 0003
TA- T-D6-553

6. AUTHOR(S)
Dr. Robert E. Fulton, Dr. Chou-pin Yeh, Dr. Karen J. Richter

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-2154

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
OUSD(A), R&AT
Rm. 3D1089, Pentagon
Washington DC 20301-3080

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This paper presents a systematic evaluation of seven data/process modeling methods for supporting the aerospace vehicle design process: three integrated Computer-Aided Manufacturing (ICAM) Definition Languages ($IDEF_0$, $IDEF_1$, $IDEF_{1X}$), Nijssen's Information Analysis Method (NIAM), the Systematic Activity Modeling Method (SAMM), the Entity-Relationship Model (ERM), and the Object-Oriented Data Model (OODM). An example of each model is given in a test application of an aircraft wing composite panel design. The conclusion reached from the study is that none of the existing modeling methodologies adequately supports the overall vehicle design process, and a recommendation is made to develop an extended information modeling methodology by combining the OODM with a process model such as $IDEF_0$ or SAMM.

14. SUBJECT TERMS
Information Model, Data Model, Process Model, aerospace design, engineering design process, database management system, DBMS, Unified Life Cycle Engineering, ULCE, Integrated Programs for Aerosapce Vehicle Design, IPAD

15. NUMBER OF PAGES
104

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

IDA PAPER P-2154

MANAGING ENGINEERING DESIGN INFORMATION

R. E. Fulton, *Georgia Institute of Technology*
Chou Pin-Yeh, *Georgia Institute of Technology*
Karen J. Richter, *IDA*

October 1989

INSTITUTE FOR DEFENSE ANALYSES

Contract MDA 903 89 C 0003
Task T-D6-553

PREFACE

This report was prepared by the Institute for Defense Analyses (IDA) for the Office of Engineering Technology, Deputy Under Secretary of Defense (Research and Advanced Technology), and the Air Force Human Resources Laboratory, Logistics and Human Factors Division, under Contract Number MDA 903 89 C 0003, Task Order T-D6-553. "Applications of Systems Engineering Techniques to Development of a Unified Life Cycle Engineering Environment."

The issuance of this report meets the specific tasks of surveying "techniques which have been used in past studies of design processes," evaluating techniques, such as "Object-Oriented Programming Techniques," and selecting "a design problem to be used as a case study in examining the applicability of the techniques."

This paper was reviewed by Drs. Jeffrey H. Grotte and Frederick R. Riddell of IDA.

CONTENTS

PREFACE iii

GLOSSARY ix

EXECUTIVE SUMMARY ES-1

A. Introduction ES-1

B. Overview ES-2

C. Conclusions and Recommendations ES-2

I. INTRODUCTION 1

A. Background 1

B. Overview 2

II. AEROSPACE VEHICLE DESIGN PROCESS 5

A. Characteristics of Aerospace Vehicle Design Process 7

B. Design Process Automation 9

C. Design Process Integration 10

1. Interface/Translation Approach 11
2. Directory Data Base Approach 13
3. Common Data Base Approach 13
4. Executive-Centered Approach 15
5. Case Studies 16

III. DATA BASE MANAGEMENT ISSUES IN THE DESIGN PROCESS 23

A. Conventional Data Base Management Systems 23

1. Hierarchical Data Model 23
2. Network Data Model 25
3. Relational Data Model 26
4. Model Comparison 27

B. Deficiencies of Conventional Data Base Models 28

1. Lack of Semantic Expressiveness 30
2. Inability to Handle Engineering Heterogeneous Data Types 30

3. Inability to Implement Dynamic Schema 30
4. Limitations of Evolvability 31

C. Design Data Base Management Systems 32

IV. DESCRIPTION OF DATA/PROCESS MODELING METHODOLOGIES 35

A. Illustration of Methodologies with Aerospace Design Application 36

B. IDEF Methodologies 44

1. $IDEF_0$ - Function Model 45
2. $IDEF_1$/$IDEF_{1X}$ (Extended) - Information Model 46
3. $IDEF_2$ - Dynamic Model 46

C. Systematic Activity Modeling Method 47

D. Nijssen's Information Analysis Method 48

E. Entity-Relationship Model 48

F. Object-Oriented Data Model 49

V. EVALUATION OF DATA/PROCESS MODELING TECHNOLOGIES 51

A. Functional Requirements 51

B. Evaluation Matrices 52

C. Additional Evaluations 61

1. Assumptions 61
2. Skills Required for Use 61
3. Scope 62
4. Graphical Representation and Software Support 62

VI. CONCLUSIONS AND RECOMMENDATIONS 63

REFERENCES 65

DISTRIBUTION LIST DL-1

LIST OF FIGURES

1. Evolution of Aerospace Vehicle Design Process 6
2. Design Network Example (IPAD Project) 8
3. Interface/Translation Approach for Design Process Integration 12
4. Translator Interface Between Application Tools 12
5. Example of Directory Data Base Approach 14
6. Design Process Integration--Common Data Base Approach 15
7. Design Process Integration--Executive-Centered Approach 16
8. Integrated Programs for Aerospace Vehicle Design Concepts 17
9. Integrated Programs for Aerospace Vehicle Design Data Base Management Concept 19
10. Integrated Design Support System Components 20
11. System Architecture for Integrated Design Support 21
12. Examples of Conventional Data Base Management Systems 24
13. Solutions to Current Data Base Management System Deficiencies 33
14. $IDEF_0$ Model for Aircraft Wing Composite Panel Design 37
15. $IDEF_{1X}$ Model for Aircraft Wing Composite Panel Design 38
16. $IDEF_2$ Model for Aircraft Wing Composite Design Panel 39
17. Systematic Activity Modeling Method for Aircraft Wing Composite Panel Design 40
18. Nijssen's Information Analysis Method Model for Aircraft Wing Composite Panel Design 41
19. Entity-Relationship Model for Aircraft Wing Composite Panel Design 42
20. Object-Oriented Data Model for Aircraft Wing Composite Panel Design 43
21. IDEF Methodologies 44
22. $IDEF_0$ Function Representation 45

LIST OF TABLES

ES-1. Comparison of the Methodologies ES-3

ES-2. Summary of $IDEF_0$ Methodology ES-4

ES-3. Summary of $IDEF_1/IDEF_{1X}$ Methodology ES-5

ES-4. Summary of $IDEF_2$ Methodology ES-6

ES-5. Summary of Systematic Activity Modeling Method ES-7

ES-6. Summary of Nijssen's Information Analysis Method ES-8

ES-7. Summary of Entity-Relationship Model ES-9

ES-8. Summary of Object-Oriented Data Model ES-10

1. Characteristics of the Conceptual, Preliminary, and Detailed Design Phases 6

2. Representative Automated Procedures for Aerospace Vehicle and Aircraft Design Process 10

3. Results of Data Model Comparison 29

4. Summary of $IDEF_0$ Methodology 53

5. Summary of $IDEF_1/IDEF_{1X}$ Methodology 54

6. Summary of $IDEF_2$ Methodology 55

7. Summary of Systematic Activity Modeling Method 56

8. Summary of Nijssen's Information Analysis Method 57

9. Summary of Entity-Relationship Model 58

10. Summary of Object-Oriented Data Model 59

11. Comparison of the Methodologies 60

GLOSSARY

AD	Activity Diagram
AFHRL	Air Force Human Resources Laboratory
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CDC	Control Data Corporation
DBMS	Data Base Management System
DDBMS	Design Data Base Management System
DoD	Department of Defense
DSS	Decision Support System
EIS	Engineering Information System
ERD	Entity-Relationship Diagram
ERM	Entity-Relationship Model
FDM	Functional Data Model
ICAM	Integrated Computer-Aided Manufacturing
IDA	Institute for Defense Analyses
IDEF	ICAM Definition Language
IDS	Integrated Design Support
IFD	Information Flow Diagram
IGES	Initial Graphics Exchange Specification
IISS	Integrated Information Support System
IPAD	Integrated Programs for Aerospace Vehicle Design
IPEX	IPAD Executive
IPIP	IPAD Information Processor
ISD	Information Structural Diagram
LOTs	Lexical Object Types

NIAM	Nijssen's Information Analysis Method
NOLOTs	Nonlexical Object Types
OODM	Object-Oriented Data Model
PAIDE	Prototype Integration Design
PDCM	Product Data Control Model
RIM	Relational Information Management System
SAM	Semantic Association Model
SAMM	Systematic Activity Modeling Method
SDM	Semantic Data Model
SDRC	Structural Dynamics Research Corporation
ULCE	Unified Life Cycle Engineering
VHSIC	Very High Speed Integrated Circuits
VLSI	Very Large Scale Integrated Circuit

EXECUTIVE SUMMARY

A. INTRODUCTION

The modern aerospace vehicle design process is a complex and sophisticated activity encompassing voluminous computations, numerous interdisciplinary interactions, and enormous quantities of data. The process requires the efforts of many individuals over an extensive period of time. The entire scope of the design process is driven by design information created during the design operations. One emerging and most critical issue in effective design is the efficient management of the engineering design data. [Ref. 1]

Studies of interdisciplinary interactions, procedure optimizations and automations, and design methodologies abound; however, little is understood about the design and development of engineering/design data bases specifically tailored to aerospace vehicle design applications. To effectively manage the design information, various computer-aided design/computer-aided manufacturing (CAD/CAM) and data base technologies and techniques, such as integrated data base concepts, distributed data base concepts, and semantic data modeling methods, have been developed and implemented.

The conventional data base management systems (DBMS), such as hierarchical, network, and relational, have been developed for business-oriented applications and do not meet data requirements for the engineering/design environment. To overcome the deficiencies of the conventional DBMS, many data/process modeling methodologies, such as Integrated Computer-Aided Manufacturing (ICAM) Definition Language (IDEF) methodologies, Nijssen's Information Analysis Method (NIAM), and Systematic Activity Modeling Method (SAMM), were developed to address well-defined information processes. Such methodologies, together with appropriate DBMSs, have been used for specific engineering tasks. These methodologies, however, have not been adequately evaluated for their relevance to the aerospace vehicle design process. The objective of this paper is to present an evaluation of the current data/process modeling methodologies with emphasis on the aerospace vehicle design application.

B. OVERVIEW

This paper presents a systematic evaluation of data/process modeling methods for supporting the aerospace vehicle design process. One of the key issues for ensuring the effective management of engineering information is the use of a DBMS specifically tailored to engineering applications. Conventional DBMSs, developed for business- and administrative-oriented environments, have been determined incapable of fulfilling the functional requirements of engineering applications. Because of the deficiencies of conventional DBMSs, many data/process modeling methodologies have been advocated and implemented for engineering applications. Such methodologies were developed to serve the needs of particular engineering tasks and, prior to this study, had not been sufficiently evaluated for relevance to aerospace vehicle design. The IDA study covered seven data/process modeling methodologies, which were evaluated by conducting a test application to an aircraft wing composite panel design. The following methodologies were evaluated with the test application:

- Three Integrated Computer-Aided Manufacturing (ICAM) Definition Languages ($IDEF_0$, $IDEF_1$/$IDEF_{1X}$, $IDEF_2$)
- SAMM
- NIAM
- Entity-Relationship Model (ERM)
- Object-Oriented Data Model (OODM).

Table ES-1 is an evaluation matrix that presents the ratings of the various features of these methodologies. Summaries of the features, assets, and liabilities of each process/data methodology are provided in Tables ES-2 through ES-8.

C. CONCLUSIONS AND RECOMMENDATIONS

The results of this evaluation indicate that none of the existing modeling methodologies can adequately support the overall aerospace vehicle design process. The OODM seems to possess many features required for the ideal design decision support system for modeling the aerospace vehicle design process. Some of the features that the OODM does not possess are embodied in other methodologies. An extended information modeling methodology, formed by combining the OODM data model with a process model (such as $IDEF_0$ or SAMM), may provide an ideal design decision support environment.

Table ES-1. Comparison of the Methodologies

FEATURE \ MODEL	$IDEF_0$/ $IDEF_{1X}$	$IDEF_1$	$IDEF_2$	SAMM	NIAM	ERM	OODM
Version Control	Poor	Poor	Poor	Poor	Average	Average	Good
Iterative and Tentative Process	Average	Poor	Average	Average	Poor	Poor	Good
Dynamic Schema	Poor	Poor	Average	Poor	Poor	Poor	Average
Design Transaction	Average	Poor	Good	Average	Poor	Poor	Poor
Modeling Style	Top-Down	Top-Down	Front-End	Top-Down	Bottom-Up	Top-Down	Top-Down
Multiple View	Poor	Poor	Poor	Poor	Poor	Poor	Average
Ease of Use	Good	Average	Poor	Good	Excellent	Excellent	Poor
Integrity Checking	Poor	Average	Good	Poor	Excellent	Good	Good
Local Constraint	Poor	Poor	Poor	Poor	Poor	Poor	Poor
Software Support	Good	Good	Poor	Good	Excellent	Excellent	Poor
User-Defined Relationship	Poor	Average	Poor	Poor	Excellent	Good	Excellent
Complex Object Modeling	Poor	Average	Poor	Poor	Average	Good	Excellent
Data Shareability	Good	Average	Poor	Good	Good	Good	Excellent
Supports Multidiscipline Team Work	Excellent	Good	Poor	Excellent	Poor	Poor	Average

Table ES-2. Summary of $IDEF_0$ Methodology

ITEM \ MODEL	$IDEF_0$
Features	• It is a functional model that describes a complex system and interrelated information/object transfer. • It provides graphics, texts, and forms that permit the system designers to quantify the existing system, propose system enhancements, and evaluate their effects in a logical way. • It strongly reinforces the top-down functional modeling approach. It gradually introduces greater levels of detail through the diagram structure of the model.
Assets	• It permits an individual to work on different aspects of the total system design yet be consistent in terms of final systems integration. • It permits complete system encapsulization in a standardized, documented form. • It permits the user to specify a complete system design to the desired level of detail . • It is a clear, concise specification methodology currently available to functionally describe total system design.
Liabilities	• Development time is too lengthy. • It is quite complex and time consuming to read. • It has only a static representation facility and cannot define the system in terms of dynamic representation. • The function names between two different modelers can be inconsistent due to their different views about the system. • Sometimes it has difficulty in pinpointing a problem area within the system.

Such a modeling method must be researched, developed, implemented, and tested to provide critically needed support of a future information-driven aerospace design process. Large scale test bed problems, such as the XV-15 tilt-rotor composite aircraft wing structure or avionics control systems, should be used to evaluate the information methodologies assessed in this report as well as any future methodology developments.

Table ES-3. Summary of $IDEF_1/IDEF_{1X}$ Methodology

ITEM \ MODEL	$IDEF_1/IDEF_{1X}$
Features	• It comprises three primary elements: -- Entities (classes of items of information) -- Attributes (classes of kinds of information) -- Classes of relations between entities. • It incorporates the necessary graphics, texts, and forms to inject an organized discipline into the process. • It provides for the measurement and control of the progressive development of the model through the routine of the modeling discipline. • It is a coherent language that supports the development of conceptual schemas.
Assets	• It produces graphical diagrams that explicitly represent data semantics. • It represents a broad range of detail, making it suitable for supporting the complete process of developing information systems. • It is independent of any DBMS and application tools. • It has been successfully applied in a variety of enterprises to achieve implementation of integrated data resources. • It provides a modularity that can protect against inaccuracy, incompleteness, inconsistency, and imprecision. • It supports disciplined, coordinated teamwork.
Liabilities	• It describes only the static behavior of information in a system. • Considerable knowledge is required for implementation, and building the data model is time consuming. • Inexperienced users often generate a non-normalized form and later cause data base anomalies.

Table ES-4. Summary of $IDEF_2$ Methodology

ITEM \ MODEL	$IDEF_2$
Features	• It describes a time-varying behavior in a systematic way such that the descriptions can be analyzed using computer simulations to generate a measure of system performance. • It decomposes into four submodels (graphic components): --Entity flow networks --Resource disposition trees --System control networks --Facility diagrams. • It models system behavior by examining the manner in which entities flow through the system and the reaction of the system to the entity flow.
Assets	• It is suitable to measure the performance in terms of time. • It can model probability of occurrence, personnel involvement, decision making, and interactions among activities and events. • It is suitable to model the dynamic behavior of bounded systems, such as manufacturing processes. • It predicts and experiments with a system's dynamic behavior without implementing and building the system. • It makes use of computer simulation techniques and reduces human error.
Liabilities	• It is difficult to understand and implement due to complexity. • It can handle only well-bounded manufacturing processes. It is not suitable to model an unbounded system, such as a design activity.

Table ES-5. Summary of Systematic Activity Modeling Method

MODEL / ITEM	SYSTEMATIC ACTIVITY MODELING METHOD (SAMM)
Features	• It provides a systematic approach by using a top-down hierarchical decomposition technique approach. • An activity diagram (AD) is used to show the interrelationships between activities by indicating data and data flow through their relationships. • It can be used to model the design networks that are the fundamental building blocks for the design process. • It is designed to be expandable to the level of detail desired by the designers.
Assets	• It allows the individual to construct the model in a parallel and modulized manner without involving the details of other activities. • It provides information such as the number of iterations, the quantity of data, and whether the activity can be performed using computers. • It permits the designer to specify a complete system design to the desired level of detail. • It permits the design to be reviewed and examined by many individuals, and comments by these individuals can be incorporated in a consistent, standardized manner. • The cost and time drivers can be quantified.
Liabilities	• It does not indicate a specific sequence or flow as evolving over time. This fact is frequently misunderstood by users. • It does not have information about the involvement of mechanisms such as design tools, computer hardware, or personnel. • It encourages the designers to concentrate on individual activity, without seeing the process as part of the entire system. • It is a static representation of the activity, which may be problematic since designers have difficulty perceiving the design process in terms of static data flow.

Table ES-6. Summary of Nijssen's Information Analysis Method

ITEM \ MODEL	NIJSSEN'S INFORMATION ANALYSIS METHOD (NIAM)
Features	• It is a binary-relationship conceptual data model. • It is a means of capturing information requirements in user-understandable terms, modeling and analyzing the requirements in a formal information model, and translating conceptual information requirements into implementable specifications. • Relationships between object types are derived through entity-joins rather than symbol-joins. • It is a rule-based modeling technique that can be easily mapped into the data base schema and data specifications up to the third normalized form using functional decomposition and an information structural diagram (ISD).
Assets	• It is easy for non-technical people to use because schemata defined in terms of the model can be read almost like a natural language. • It supports a variety of constraints that are not available in the conventional data models. • Users have complete freedom to override the form suggested by NIAM and dilute the high level of normality. • It uses a semantic binary association between objects in generalized object classes; therefore, it is capable of modeling any environment. • Information can be easily automated by the computer algorithms to transform the conceptual schema into a logical data base schema.
Liabilities	• It is not considered a "real" data model since it does not provide a nice and well-defined set of data manipulation operations. • It does not provide capacities for view definition. • It is inefficient even with simple queries, requiring a greater number of joint operations than conventional data models.

Table ES-7. Summary of Entity-Relationship Model

MODEL / ITEM	ENTITY-RELATIONSHIP MODEL (ERM)
Features	• It is one of the earliest conceptual data models. • It supports the top-down approach. • It consists of three basic constructs: entities, relationships, and attributes. • It can model composite entities or their relationships. • The Entity-Relationship Diagram (ERD) provides users a visual immediacy that makes ERM a popular conceptual data model.
Assets	• The ERM's basic construct is very simple to represent and learn. • The ERD is a comprehensive and simple diagrammatic technique. • Many-to-many relationships are easy to implement. • ERD can be easily mapped into a relational data base structure with up to the third normal form. • It is supported by the well-developed entity relationship modeling tools.
Liabilities	• It assumes that an entity can be represented by a single relation. • Even if classified as a semantic data model, ERM still cannot provide sufficient semantics for engineering design objects. • It provides the modelers with a great deal of freedom to model the enterprise; hence, models generated by different individuals can have many discrepancies.

Table ES-8. Summary of Object-Oriented Data Model

MODEL / ITEM	OBJECT-ORIENTED DATA MODEL (OODM)
Features	• It models all of the conceptual entities with a single concept--objects. • Each object encapsulates data and procedures to operate on the data. • It has four characteristics: data abstraction, information hiding, inheritance, and dynamic binding. • It provides a hierarchy of types of objects and the ability to inherit the properties of the parent object types. • It allows application programs to view a class of abstract data objects completely in terms of a set of characterizing operations.
Assets	• Complex design entities can be represented more directly, with less encoding, meaning fewer levels of indirection. • It offers fast response in design applications. • Update operations and constraints are an integral part of the data base. • Data independency is maintained. • An efficient programming language interface can be developed. • Iterative and tentative nature of design is supported. • Multistage nature of design is supported. • Dynamic schema and data base operations are extendable. • Data can be shareable. • Versions, alternatives, and revisions can be easily implemented.
Liabilities	• The concept is difficult to implement. • The dynamic binding mechanism has high run-time costs. • A variety of the object-oriented paradigms, each defining different terminologies and meanings, cause inconsistencies and confusion to designers not proficient in DBMS.

I. INTRODUCTION

The modern aerospace vehicle design process is a complex and sophisticated activity encompassing voluminous computations, a lengthy time span, numerous interdisciplinary interactions, and enormous quantities of data. An emerging and most critical issue in effective design is the efficient management of design data. [Ref. 1]

Studies of interdisciplinary interactions, procedure optimizations and automations, and design methodologies abound; however, little is understood about the design and development of engineering/design data bases tailored to aerospace vehicle design applications. The conventional data base management systems (DBMS), such as hierarchical, network, and relational, were developed with business-oriented applications in mind and do not meet data requirements for the engineering/design environment. To overcome the deficiencies of the conventional DBMS, many data/process modeling methodologies, such as Integrated Computer-Aided Manufacturing (ICAM) Definition Language (IDEF) methodologies, Nijssen's Information Analysis Method (NIAM), and Systematic Activity Modeling Method (SAMM), were developed to address well-defined information processes. Such methodologies, together with appropriate DBMSs, have been used for specific engineering tasks. These methodologies, however, have not been adequately evaluated for their relevance to the aerospace vehicle design process. The objective of this paper is to present a systematic and complete evaluation of the current data/process modeling methodologies, with an emphasis on an aerospace vehicle design application.

A. BACKGROUND

In FY1987, the Institute for Defense Analyses (IDA) conducted two studies for the Air Force's Unified Life Cycle Engineering (ULCE) program. These studies were supported by the Air Force Human Resources Laboratory (AFHRL), Logistics and Human Factors Division, Wright-Patterson Air Force Base in Ohio. One study was an investigation of the decision support requirements for an ULCE environment. To identify research priorities for decision support in ULCE, a working group composed of members of academia, industry, and government was convened. The group identified the following actions as necessary for implementation of a decision support system (DSS) in ULCE:

> Engineering needs for data base design must be expected to differ from those that spring from DSS structure. It is important to identify those aspects of DSS data base design that lead to competition with, contradiction of, or require extension to data bases that are engineering dictated. The advantages and disadvantages of advanced data base technology, such as object-oriented data bases versus the relational data base structure more commonly used in business DSS, must be examined. [Ref. 2]

The second IDA study focused on the architecture and integration requirements for an ULCE design environment. A recommendation resulting from this study was that "research should be conducted in the areas of data base management systems, data modeling, and applications of object-oriented techniques to design systems." In particular,

> An extensible representation of design engineering data, which incorporates features of the network, hierarchical, relational, and object-oriented data model types, is needed for ULCE. This data model will provide a foundation for the integration of the ULCE design tools and the knowledge/data base management systems. [Ref. 3]

To address some of the problems posed in these studies, the AFHRL, Logistics and Human Factors Division, funded additional IDA research during FY1988. IDA enlisted the expertise of Dr. Robert Fulton, Professor of Mechanical Engineering at Georgia Institute of Technology, and his graduate student, Chou-pin Yeh. Dr. Fulton was Program Manager of the Integrated Programs for Aerospace Vehicle Design (IPAD) at NASA-Langley Research Center from 1972 to 1984. IPAD was one of the most extensive investigations of the aerospace design process and resulted in comprehensive models of that process. Chou-pin Yeh also has design experience and is working with Dr. Fulton to identify the relevance of information modeling methods to engineering design tasks. This report is the result of their work, performed for IDA, in identifying and evaluating the data and process modeling methods for aerospace design.

B. OVERVIEW

This paper begins with a description of the aerospace vehicle design process. The characteristics of the design process are identified in Section II-A. The automation and integration of the design process are addressed in II-B and II-C, respectively. Chapter III contains a description of the conventional data base models, their defects, and why a design data base management system is needed to effectively manage design information. The current data/process modeling methodologies are described in Chapter IV and evaluated in Chapter V. Conclusions and recommendations for further research are provided in Chapter VI, including ideas for an extended data/process modeling methodology that meets the functional requirements for an aerospace vehicle design environment.

II. AEROSPACE VEHICLE DESIGN PROCESS

Design is the synthesis of related activities, including the design of products, processes, manufacturing systems, software, and organizations [Ref. 4]. Engineering design is the result of rational decision making and has been defined as a mapping process through which needs are translated into functional requirements and then into a product. Engineering design can be viewed as a process of decomposing and refining interrelated representations of functions, structures, and behaviors of the end product and its components. The design process describes the gathering, handling, and creative organizing of information relevant to the design practices [Ref. 5].

The modern aerospace vehicle is a complex integration of sophisticated technical systems manufactured to the exact standards required for safety, economy, and mission performance [Ref. 6]. The aerospace vehicle design process passes through several phases as it progresses from an initial conceptual design to the final detailed design [Ref. 7]. As depicted in Figure 1, aerospace vehicle design is an evolutionary process. Research, development, and marketing activities result in new design concepts and technologies. The ideas generated from these new concepts and technologies enter a conceptual design phase, where the design characteristics are scoped to allow progression to the preliminary design phase. When the design is sufficiently mature, it is authorized to proceed to the detailed design phase. (See Table 2 for some of the design characteristics of the conceptual, preliminary, and detailed design phases.) After the detailed design phase, the product is manufactured, tested, and delivered. Design support for the product in production must be a continuing activity to cover the changes and modifications in the future product improvement [Ref. 7]. This total process involves many subtasks and cycles in a variety of sequences over a lengthy time span. The phases of aerospace vehicle design do not occur in a simple linear progression; there is a great deal of overlap, and the activities of the phases often occur in parallel. Schedules are very limited in the aerospace industry, and design operations cannot be delayed until a prior task is completed [Ref. 9].

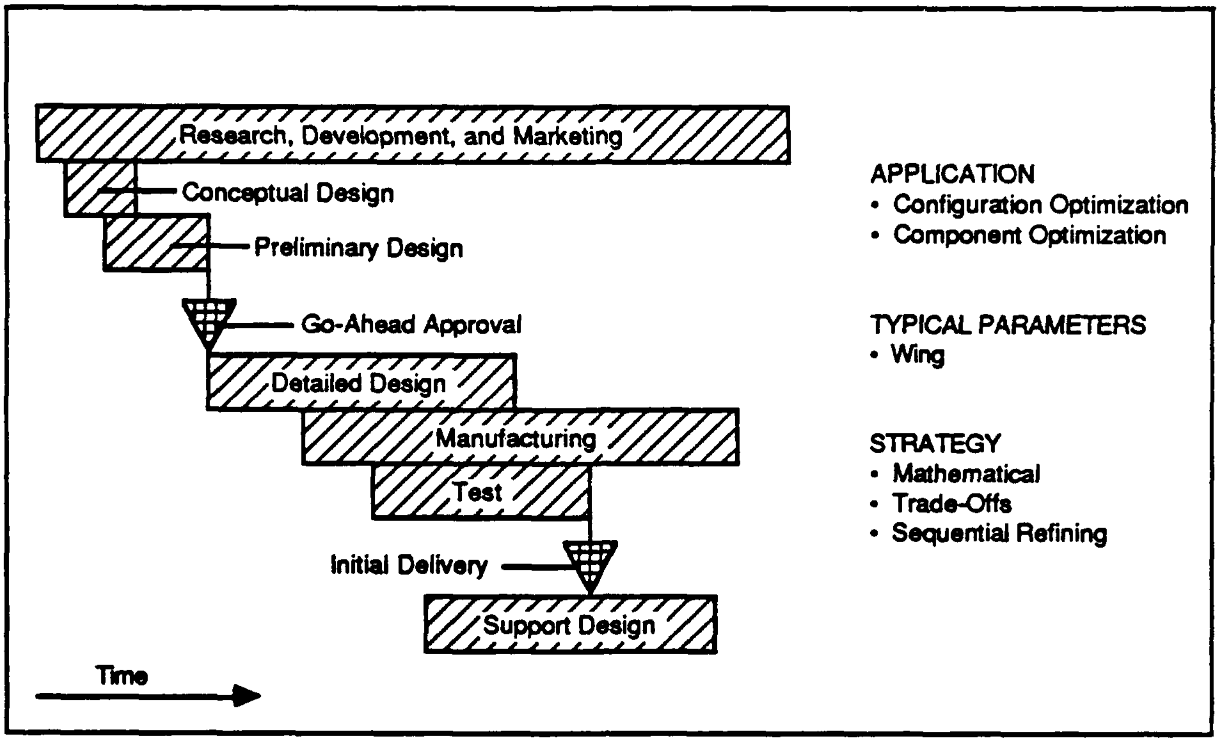


Figure 1. Evolution of Aerospace Vehicle Design Process

Table 1. Characteristics of the Conceptual, Preliminary, and Detailed Design Phases

DESIGN PHASE / CHARACTERISTIC	CONCEPTUAL	PRELIMINARY	DETAILED
Manpower	20	50-500	300-3,000
Flowtime	2-3 weeks	6 monthss	1-5 years
Configurations Examined	100s	10s	1
Weight Accuracy	<92%	92-98%	>98%
Objectives	Define market potential	Select acceptable vehicle	Prepare manufacturing and test plans
Output	Concept, assumptions, and philosophy	Configurations, specifications, and approaches	Shop data, parts, facts, and drawings

Basically, design is an exploratory process during which abstraction is employed to simplify the design process. The process of abstraction results in a design hierarchy. In aerospace vehicle design, a design network is a useful representation of the design hierarchy, to identify and describe the logical information flow for any level of the design process [Ref. 9] (see Figure 2).

A. CHARACTERISTICS OF AEROSPACE VEHICLE DESIGN PROCESS

Aerospace vehicle design is an unusual system process because it has special features not found in other complex processes. By carefully examining and studying the design process, the characteristics of aerospace vehicle design can be identified as follows:

- *Complex and Sophisticated.* The modern aerospace vehicle is an integration of complex geometry, advanced technology, intricate manufacturing processes, and sophisticated business strategies. The structure of an aerospace vehicle contains a vast number of parts and details. For instance, the airframe of a wide-body jet transport contains more than one million parts [Ref. 7].
- *Time Consuming and Expensive.* Aerospace vehicle design involves hundreds of individuals, and completing the entire design process can take several years. As a result, the cost of design can be enormous [Refs. 7, 9].
- *Requiring Many Supporting Design Tools.* It requires numerous design tools and aids, such as analysis, optimization, and geometrical modeling packages to support and facilitate the design process.
- *Creating Large Quantities of Data.* Due to the considerable size and complexity of an aerospace vehicle, its design creates large quantities of data. The data include design information (such as functional or graphical descriptions of design objects), information to validate designs (test cases and simulation results), and information that documents the design process [Ref. 10].
- *Dynamic.* Aerospace vehicle design is a highly creative, newly developed, and informal technology rather than a stereotyped, standardized, and well-established process. Hence, the design definition often changes to rectify design difficulties, accommodate new technology, and incorporate modified design criteria, making the aerospace vehicle design process a very dynamic environment.
- *Iterative.* The design process is not a logically progressive path. Instead of iterating the same algorithm time and time again, any one solution may require a number of iterations at various stages of the process until the design criteria and specifications are met [Ref. 11]. Resizing the configuration of an aircraft

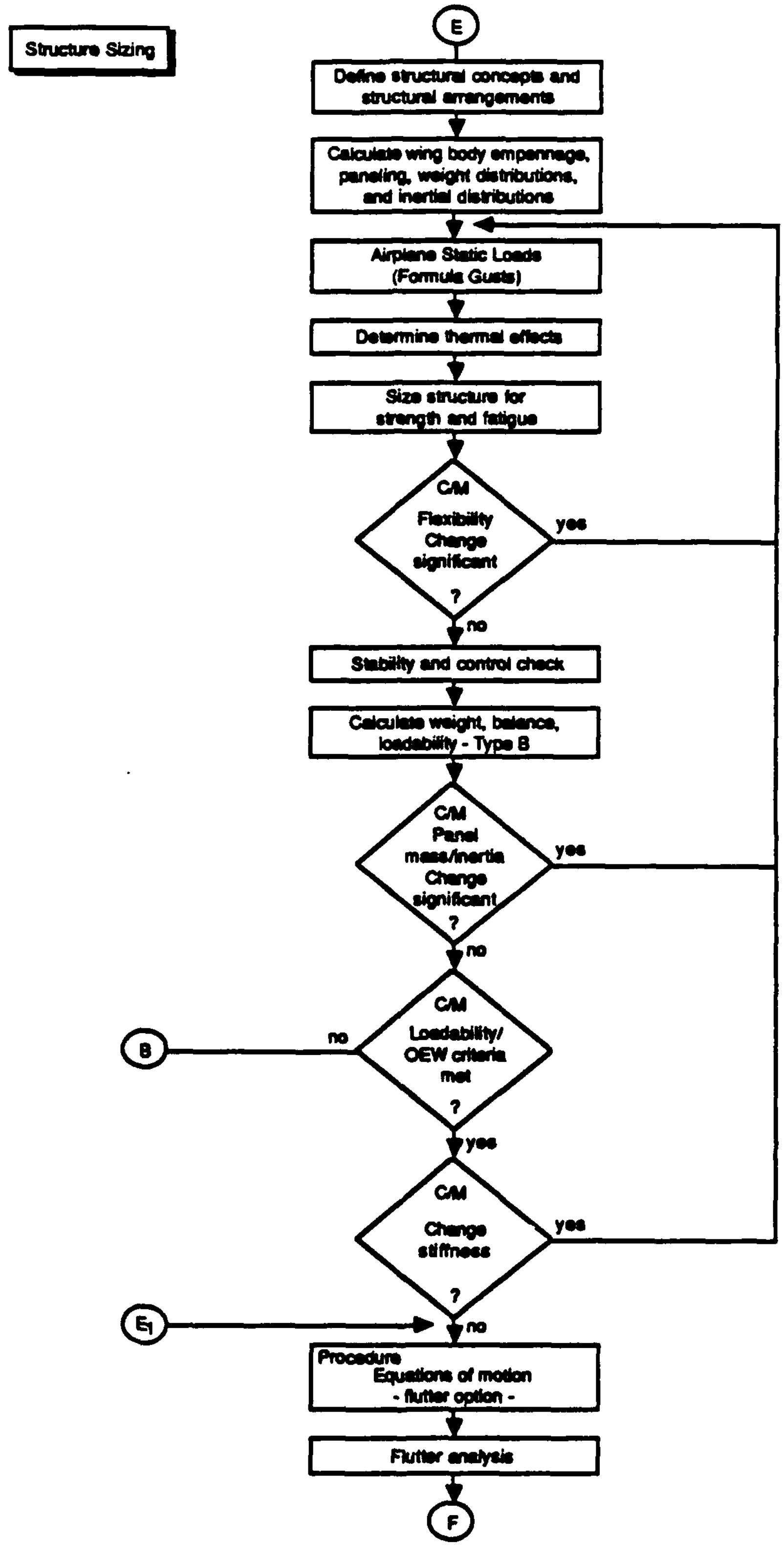


Figure 2. Design Network Example (IPAD Project)

or its parts, for example, may take hundreds of iterations and millions of computation steps to achieve the optimal weight-strength design [Refs. 7, 8, 9].

- *Work Done in Parallel.* Due to the tight schedules of aerospace vehicle design activities, the design tasks are often decomposed into subtasks, which are accomplished by different teams working in parallel.
- *Numerous Interactions Between Designers.* Since design activities are performed in parallel, many tasks overlap, causing much interaction and information transfer among designers [Ref. 9].
- *Incorporating Optimization Procedures in the Early Design Stage.* In aerospace vehicle design, the optimization techniques play an important role since one design objective is to attain the maximum or minimum value of some merit functions. Structural optimization can determine a minimum weight design of a structure subjected to constraints on design requirements. The optimization procedures are, in general, useful in conceptual and initial preliminary design phases, since the number of variables and constraints are small enough to characterize the entire system [Ref. 7].
- *Multidisciplinary.* The design process encompasses all activities required to generate the data needed to produce a product; therefore, it covers a wide scope of technical disciplines ranging, for example, from aerodynamics to structures to manufacturing to economics [Refs. 7, 8, 9].

B. DESIGN PROCESS AUTOMATION

Prior to the 1970s, the aerospace vehicle design synthesis was performed manually. The designers synthesized the production definition and its realization as a certified manufactured entity through the use of abstractly represented information, such as engineering drawings, bills of materials, and analysis results. This "paper" method involved extensive human activities, which were often error prone and time consuming [Refs. 9, 12]. In addition, due to human limitations in dealing with such a great volume of complex information, the technical depth was restricted and was not adequately maintained in some phases of the design process [Ref. 7]. Therefore, human productivity needed to be enhanced through computer assistance, to reduce time spent on routine functions and add greater technical depth and optimization in the early stages of design, when basic concepts are selected. Since the eariy 1970s, many computerized automated programs have been developed for use in the aerospace vehicle and aircraft design process (see Table 2). In general, these application programs provide capacity in one design level and are

Table 2. Representative Automated Procedures for Aerospace Vehicle and Aircraft Design Process

DISCIPLINE	PROGRAM NAME	ACRONYM
Vehicle Synthesis	General Aviation Synthesis Program Aerospace Vehicle Interactive Program Configuration Development System	GASP AVID CDS
Aerodynamics	Dynamic Loads Analysis of Flexible Airplanes Aerodynamic Panel Loads Program	DYLOFLEX USSAERO
Structural Analysis	NASA Structural Analysis Program Structural and Optimization Program Automated Program for Aircraft Structure	NASTRAN ACCESS III APAS III
Optimization	General-Purpose Optimization Program Flutter and Strength Optimization Program	OPT FASTOP
Structural Sizing	Aircraft Sizing and Performance Program Vehicle Sizing and Performance Evaluation Program	VASCOMP II VSPEP
Propulsion and Power	Computation of Three-Dimensional Combustor Performance Program	COM3D
Mission Analysis	Goddard Mission Analysis System	GMAS
Geometric Modeling	Aircraft Geometry Generator Helicopter Geometry Modeler	GEMPAK HESCAD

unidisciplinary. Each has its own language and means for representing information, thus making it almost impossible for cooperating design groups to share information [Refs. 7, 13]. This results in an environment composed of what are termed *islands of automation*. Overall performance using this approach is not satisfactory [Ref. 14]; the generation of computer-aided design/computer-aided manufacturing (CAD/CAM) programs provided design tools for partial automation of the total design process without a focus on integration of functions as a primary goal [Ref. 15].

C. DESIGN PROCESS INTEGRATION

Effective management of information in the engineering process is key to efficiently managing the design process [Ref. 1]. To aid the designers in managing engineering information, CAD/CAM technology has been widely developed and implemented [Ref. 16]. CAD/CAM applications have enhanced the design process by providing the design tools or application programs for generating a new design or modifying an existing similar design.

When CAD/CAM systems were first introduced, each operated independently, and design data were prepared manually as input to the individual application programs containing their own local data bases. Due to these islands of automation, enormous overhead costs were incurred in managing design data [Ref. 13]. To consistently manage design and analysis of engineering objects, data in engineering applications must be processed in an integrated manner. An integrated CAD/CAM system provides for the greatest interaction and flexibility in program utilization and the highest potential for automation without losing insight and innovation. In addition, an integrated system can provide for an intelligent dialogue between the designer and the computer so that they may augment and complement each other in managing and accomplishing the design task.

There are four basic approaches for integrating CAD/CAM systems, which are discussed in the following sections.

1. Interface/Translation Approach

The first approach consists of constructing an interfacing network among various CAD/CAM systems so that the output of the upstream program automatically becomes the input of the downstream program [Ref. 16] (Figure 3). This approach, however, creates several problems:

- It requires many translators, one between every two CAD/CAM systems, to handle the translations (input/output data conversion) and the communication tasks. If there are N different CAD/CAM systems, then $N^*(N-1)/2$ translators are required (Figure 4) [Refs. 13, 17].
- If one system or tool changes its status, all other translators related to this system must adjust their status accordingly. This makes the entire integrated CAD/CAM system very rigid and difficult to modify and maintain, especially when there are many systems [Ref. 18].
- Translators operate in one direction only and do not provide the degree of interactivity required for normal decision making.
- Translations and interfaces are simply too time-consuming and costly to execute and maintain for these purposes.

To streamline the translation and interface, several data exchange standards such as the Initial Graphics Fxchange Specification (IGES) have been developed and do somewhat improve this approach; however, difficulties in flexibility and interactivity still exist.

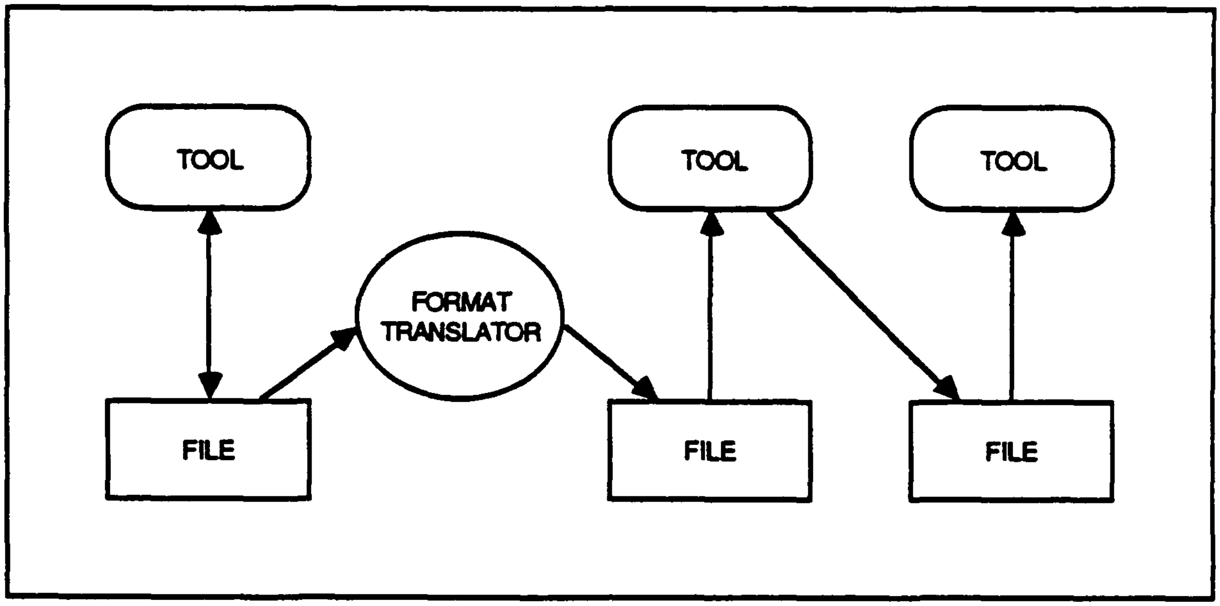


Figure 3. Interface/Translation Approach for Design Process Integration

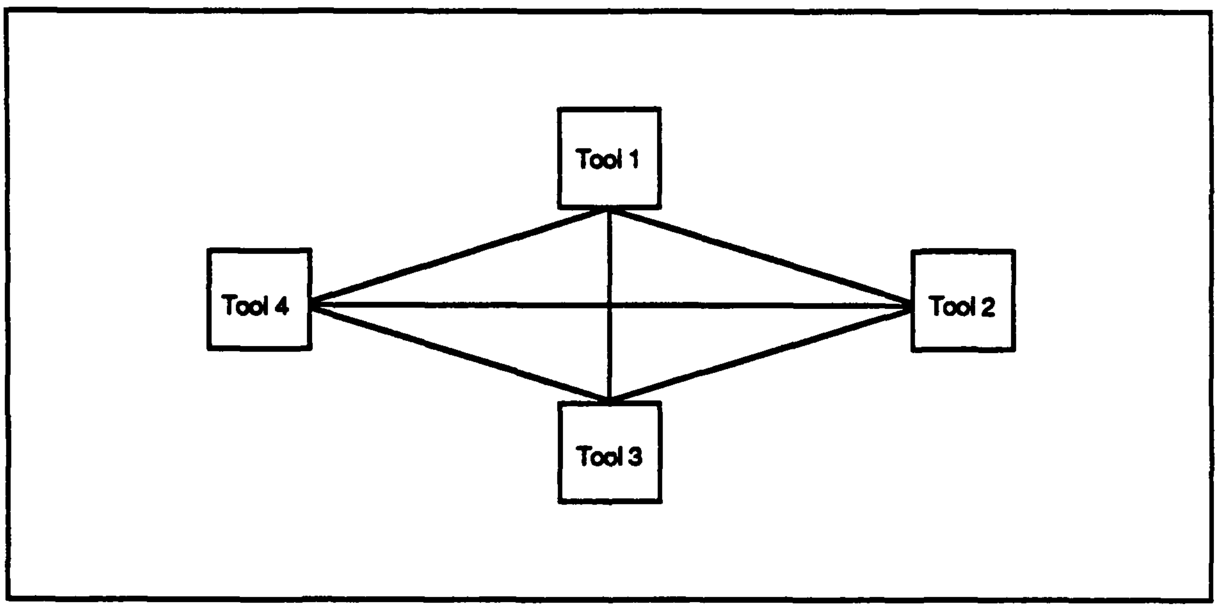


Figure 4. Translator Interface Between Application Tools

2. Directory Data Base Approach

The directory approach uses a data base with the traditional librarian functions to track design and manufacturing information, in computer files or on paper [Ref. 17] (Figure 5). The data base structure maps engineering document names into file names and locations and allows the design engineers to ask for design documents without having to recall file names. This approach is easy to implement because neither the file formats nor the applications are changed from their original form. In addition, by providing the necessary support procedures, most functions are transparent to the users. However, this approach does have its drawbacks:

- It provides minimal assistance to the designers since they must still specify the document type and the part desired.
- The computer is used to manage engineering data files rather than individual fields and records as it usually does in data base management systems. File access is sequential, whereas records are randomly accessed--data is accessed more quickly by the latter method.

3. Common Data Base Approach

The third approach to integrating CAD/CAM systems eliminates the limitations of the first two, by achieving integration at the data base level--by connecting all systems to a common data base. From the user's view, the communication or interface between any two systems always occurs through the common data base [Refs. 13, 17, 19] (Figure 6); however, the common data base can be centralized or distributed among various locations throughout the systems. The common data base approach best provides the full benefits of data base management technology. The common data base managed by a powerful commercial DBMS provides the enterprise with a flexible design/manufacturing environment because the data base is easily extended to support additional applications when needed. These commercial DBMSs, called conventional DBMSs, embody many customized features for easier application development, simplified application maintenance, improved data shareability, and redundancy.

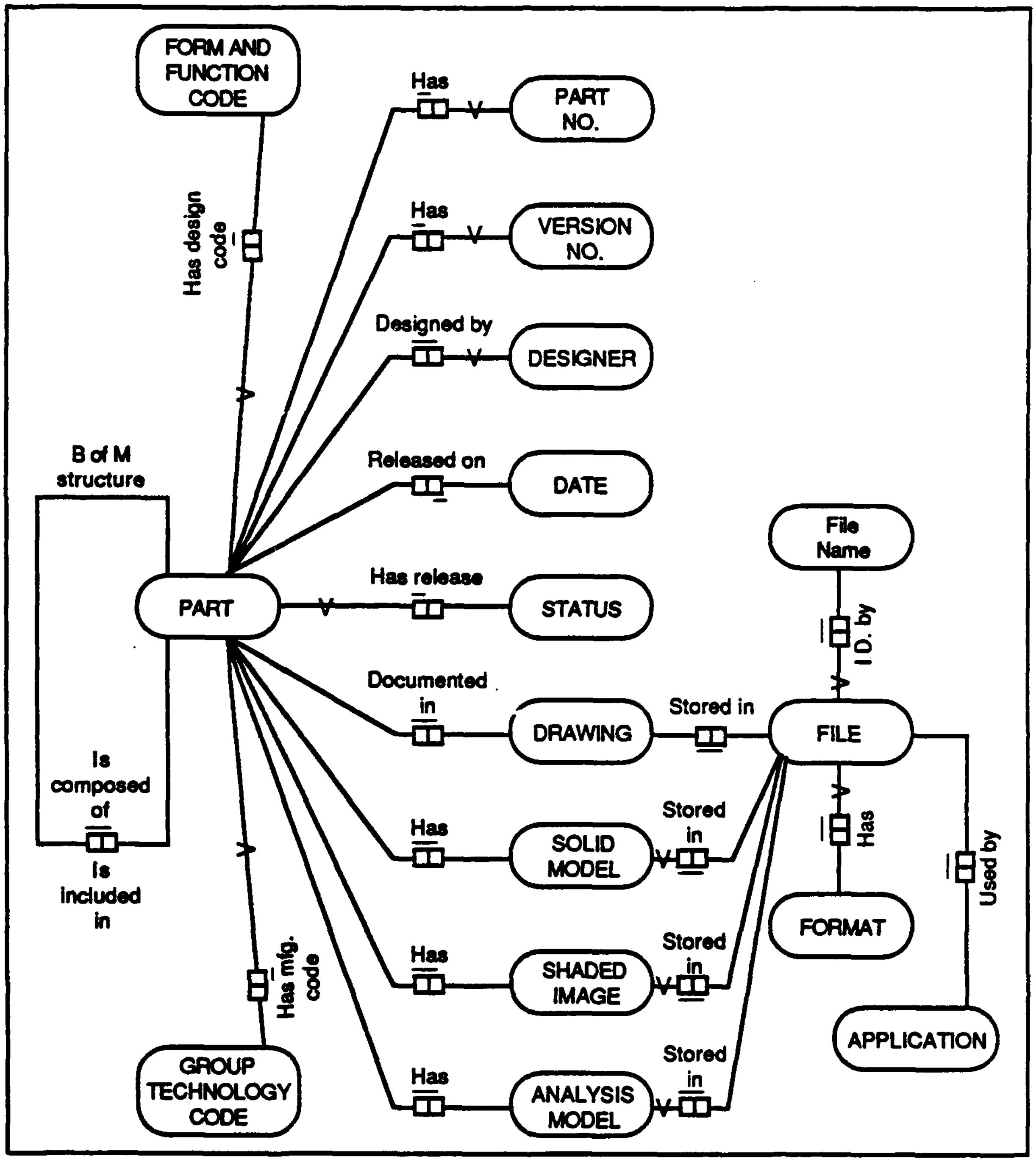


Figure 5. Example of Directory Data Base Approach

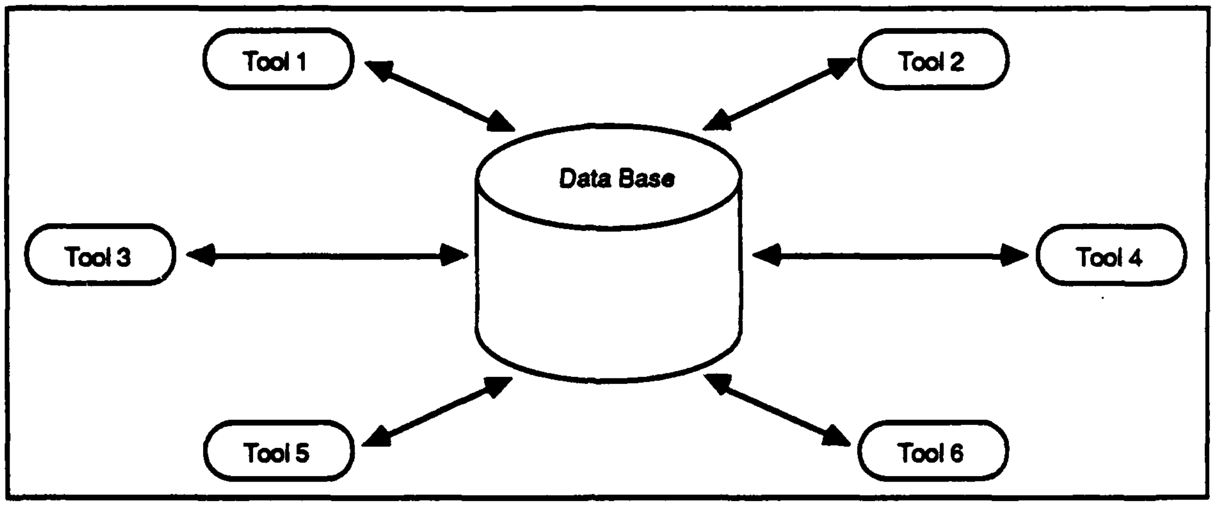


Figure 6. Design Process Integration--Common Data Base Approach

Integration of data bases into a centralized common data base may pose some disadvantages:

- Because the data from individual data bases are integrated in a single data base, the sense of ownership and the responsibility for data are easily lost [Ref. 18]. As a result, inaccurate data may not be detected.
- An integrated common data base may also threaten privacy. In an integrated common data base environment, it is easier to gain access to classified information.
- This approach requires extensive efforts to convert existing CAD/CAM applications to a general baseline data base structure because most of the current CAD/CAM application tools are not designed for the data base approach.
- The common data base approach is designed to support a global view of data and tends to suppress the local views of each application; in some cases this may not be desirable [Ref. 18].

4. Executive-Centered Approach

The executive-centered architecture differs from the common data base approach in that in addition to the common data between application programs, the application programs provide for communication between the program and the user [Refs. 17, 18]. Programs are also properly synchronized by an executive to ensure overall system efficiency.

The executive-centered architecture consists of four key components (see Figure 7).

- A data base
- A user interface
- Application programs
- An executive.

The role of the data base under this approach is not reduced in its importance although it does not play the central role that the executive does.

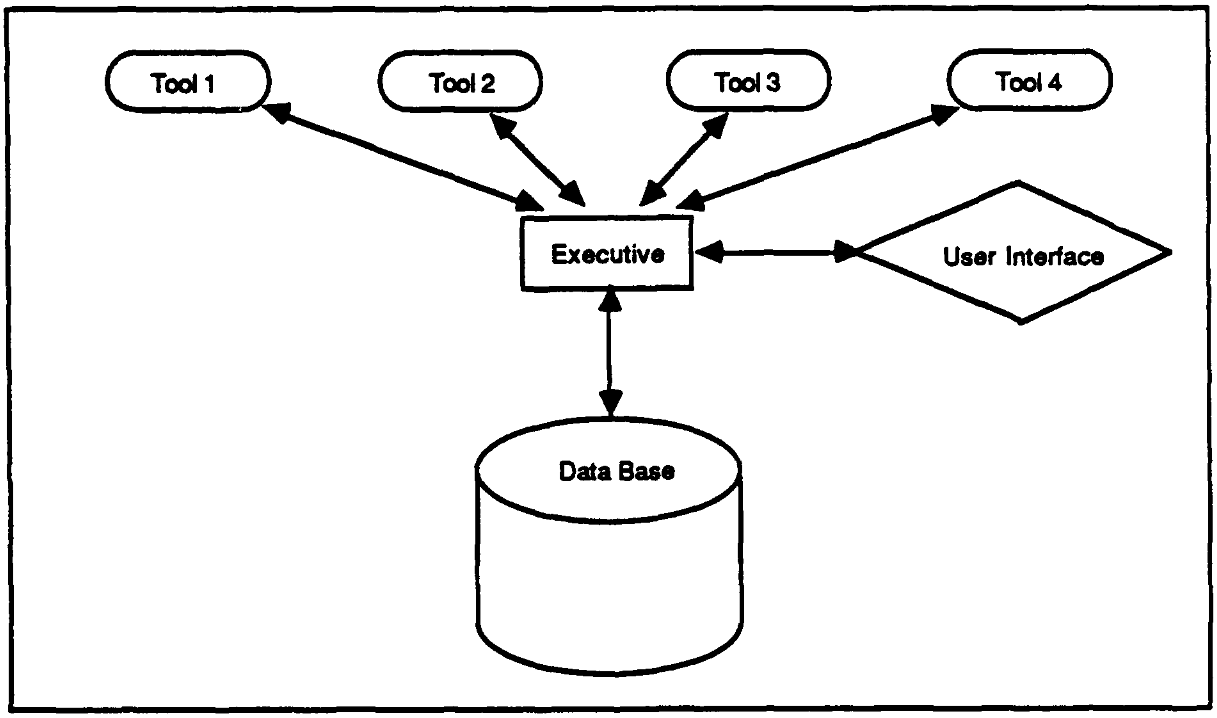


Figure 7. Design Process Integration--Executive-Centered Approach

5. Case Studies

To explore and solve this integration problem, many projects using conventional DBMS to manage data have been initiated and sponsored by the government and private industry [Ref. 8]. Several representative projects are briefly described in the following paragraphs.

a. Integrated Programs for Aerospace Vehicle Design

IPAD (Figure 8), initiated by the Boeing Company under contract to NASA-Langley Research Center in the early 1970s, was the first major project focusing on the integration of aircraft and aerospace vehicle design, analysis, and manufacturing [Refs. 16, 21-25]. One of the primary goals of the IPAD project was to increase designer productivity through the use of system software and design methods that augment technical capability and creativity, while reducing cost and flow time. The IPAD project has considered applying data base management technology to all phases of the product life cycle: CAD, CAM, and operations, and maintenance. IPAD also investigated the application of data base management technology through a common data base management facility. IPAD developed extensive documentation of the aerospace vehicle design process and information.

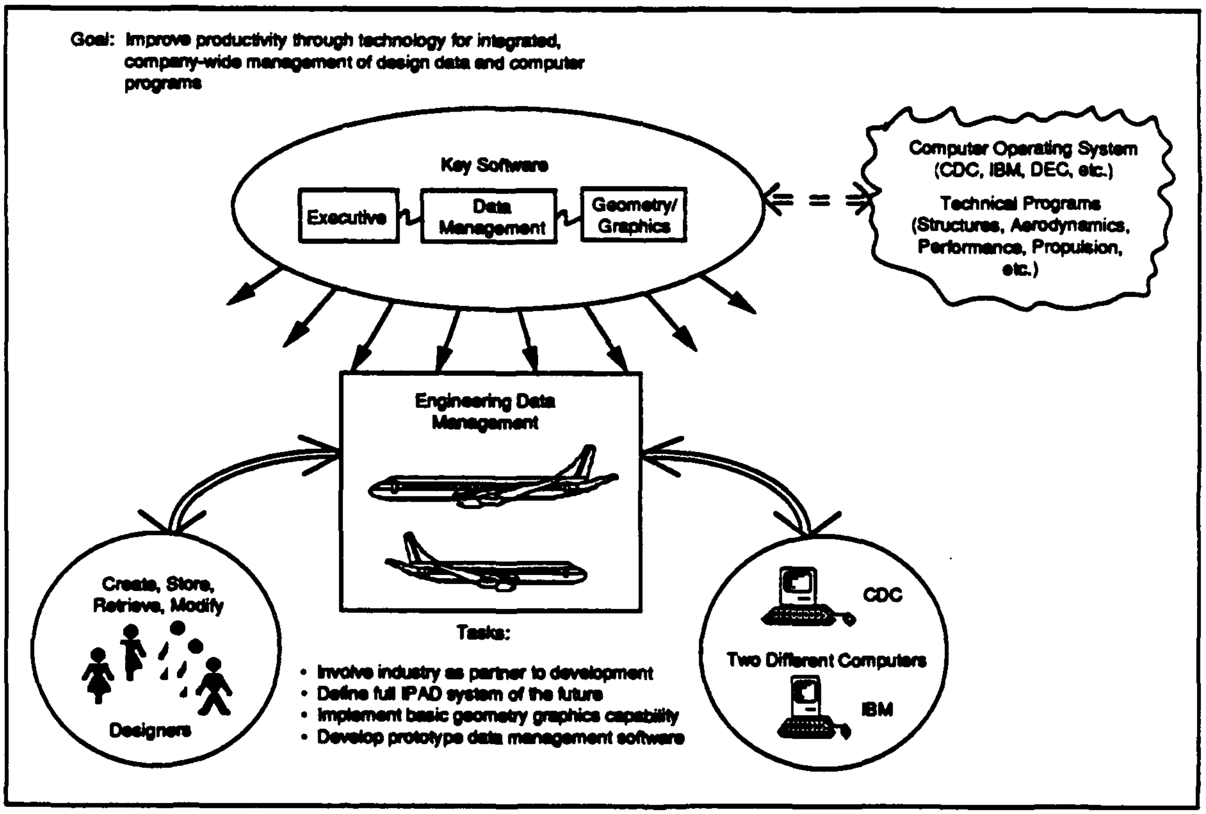


Figure 8. Integrated Programs for Aerospace Vehicle Design Concepts

IPAD is composed of the following six key components [Ref. 25]:

- *IPAD Executive (IPEX).* IPEX provides a standard set of services to the other key components, isolating them from the host computing system and providing internal functions and data needed for distribution of IPAD functions, data, and application programs. The standard services include process control, input, output, file operations, interprocess communication, and access to certain host resources and services.
- *System Functions.* The system functions provide internal services to IPAD functions, such as the collection of performance data, putting such data in files for later processing, and collecting user data.
- *User Interface.* The user interface handles all of the dialogue between the system and the user. Through IPEX, it initiates the user functions and application programs and obtains data management services from the data management system. It also reads and interprets system languages, executes executive-level commands, and aids the user in handling certain abnormal situations.
- *IPAD Data Management System.* IPAD has developed a prototype DBMS at both the local and global levels. A system denoted Relational Information Management (RIM) was developed for local-level data management. RIM, based on a relational data model, has many features such as interactive queries, report writer, and FORTRAN interfaces. IPAD also developed a global DBMS called the IPAD Information Processor (IPIP), a multiuser DBMS supporting multimodels (relational, hierarchical, and network). IPIP employs a multischema architecture to allow different users to have their own views of the same physical data.
- *User Function.* The IPAD user functions provide support to the design process in project management, design, training, data definition, data manipulation, query, pre-compilation, application program development, and document preparation.
- *Application Programs.* IPAD provides a standard user interface so that the users can install their own application programs and integrate them into the system.

The principal advantage of the IPAD approach is that the DBMS can support unified description, manipulation, and management of the data of the organization. The result is a reduction in the duplication of design data, which minimizes problems in maintenance of data base consistency and update efficiency. IPIP generated the concepts for both distributed and shared data bases. The distributed data base allows communications among the existing softwares to be dispersed geographically in heterogeneous computer hardwares (Figure 9). The concept of shared data base provides a common interface, which aids the integration of various design activities and computer-based support systems. While performing its information storage and retrieval functions, the system can also maintain data base integrity and enforce organization security rules [Ref. 26].

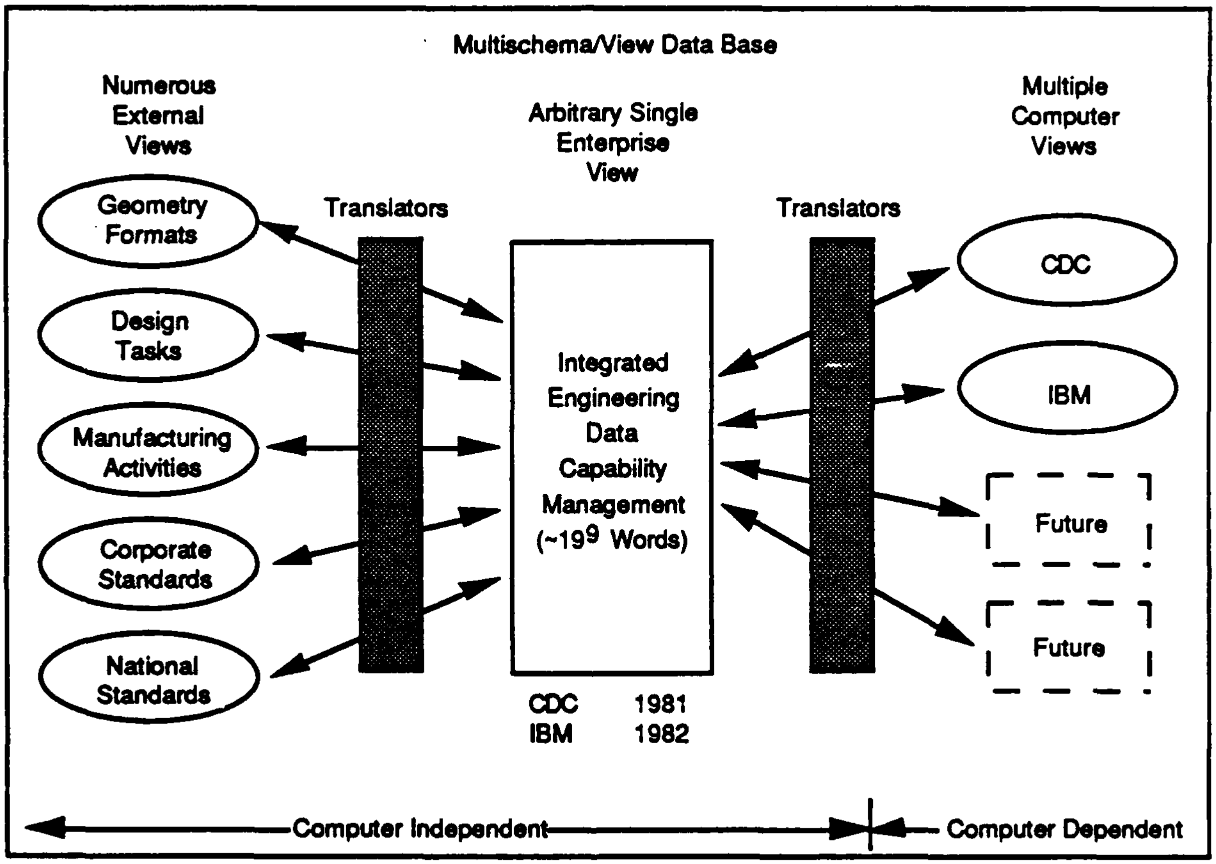


Figure 9. Integrated Program for Aerospace Vehicle Design Data Base Management Concept

To illustrate the IPAD concept and to aid instruction on integration concepts, the Prototype Integration Design (PRIDE) system was built. The system primarily focuses on structural analysis but can be readily expanded to accommodate other capacities.

b. Integrated Design Support System

The Air Force's Integrated Design Support (IDS) System is an integrated technology program that captures the critical technical engineering information necessary to perform the functions of maintenance, modification, repair, and reprocurement of the weapon systems [Refs. 19, 27] (Figure 10). IDS uses an integrated data base so that

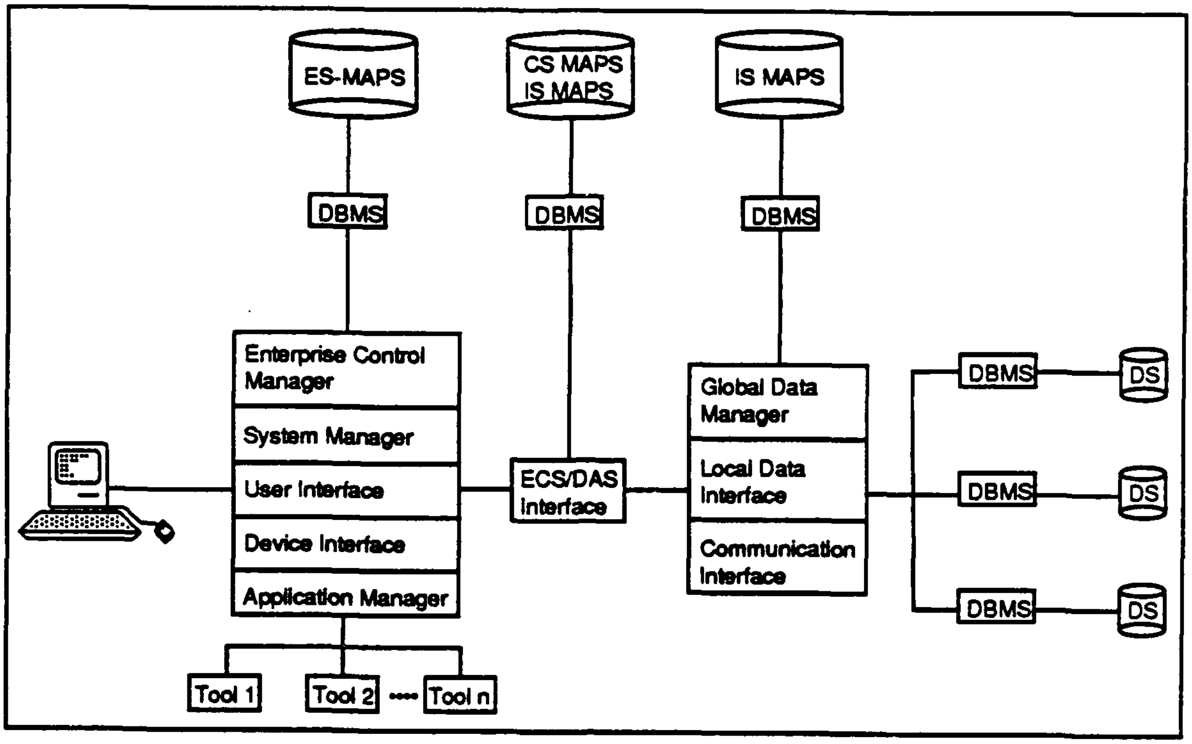


Figure 10. Integrated Design Support System Components

information can be shared by each discipline throughout the product life cycle. IDS applies several structured methodologies (IDEF) to define and understand the environment, define the data, model the dynamic behavior of the program, and predict the cost incurred.

The IDS system architecture consists of three fundamental parts: an information architecture supporting multiuser views of IDS, a computer systems architecture representing the physical view of IDS, such as different types and levels of computer hardware and software systems, and a control architecture representing the management view of IDS, including standards, procedures, data models to maintain alignment between information, and computer systems architecture. This tri-architectural conceptualization of IDS is shown in Figure 11. IDS also incorporates a mechanism, the Product Data Control Model (PDCM), for defining and controlling the technical data that can be hosted on heterogeneous computer systems and used by various users. Development of the IDS integration concept will provide the basis for a substantial improvement in the management of the technical information for both emerging and future military weapon systems. IDS will also provide the basis for life cycle cost reductions on future weapon systems.

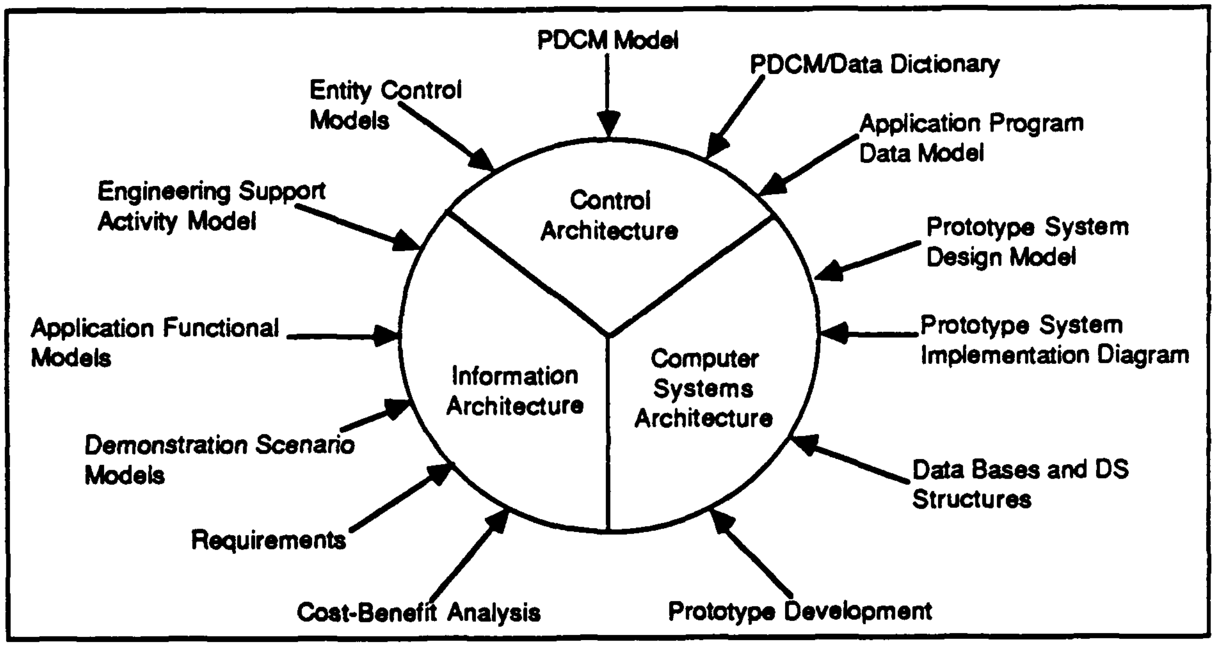


Figure 11. System Architecture for Integrated Design Support

c. Integrated Information Support System

Sponsored by the US Air Force's Integrated Computer-Aided Manufacturing (ICAM) program, the Integrated Information Support System (IISS) project is conducted by Boeing, DACOM, Structural Dynamics Research Corporation (SDRC), and Control Data Corporation (CDC) [Ref. 28]. The IISS project provides the enabling technology to logically and physically integrate a network of heterogeneous pre-existing computer hardware and software in a distributed environment. IISS has developed an integrated system that insulates the user from having to know which subsystem the user's data resides on. Using ICAM's IDEF_{1X} modeling technique, IISS focuses on the capture, management, and use of a single semantic definition of the data resource, referred to as a conceptual schema. The short-term goals of IISS are to allow data shareability and to provide a means for improving data quality and independence. The long-term goal is to provide an environment that makes all of the computers appear as one integrated computer, with all of the data seeming to reside in one data base accessed by a single, consistent type of terminal interface.

d. Engineering Information Systems

The Engineering Information System (EIS), sponsored by the Department of Defense (DoD)/Air Force's Very High Speed Integrated Circuits (VHSIC) program, is being developed using an object-oriented approach [Refs. 13, 29, 30]. Using this method, the user can define new global or local object classes, such as three-dimensional parts with operations such as display, rotate, and calculate volume. These capabilities exceed those of current DBMSs. EIS is focused on the information processing needs of engineers, managers, and administrators in the organizations involved in integrated circuit design and in the development of the tools that support the design process. EIS supports the Engineering Information Model, which provides a graphical representation of the semantics of the information in the engineering environment in which EIS operates.

III. DATA BASE MANAGEMENT ISSUES IN THE DESIGN PROCESS

The aerospace vehicle design process creates large quantities of data. A DBMS, a set of software that defines, retrieves, and modifies data stored in a data base, can be used to store and effectively manage the information, thereby increasing the productivity of the aerospace vehicle design operations.

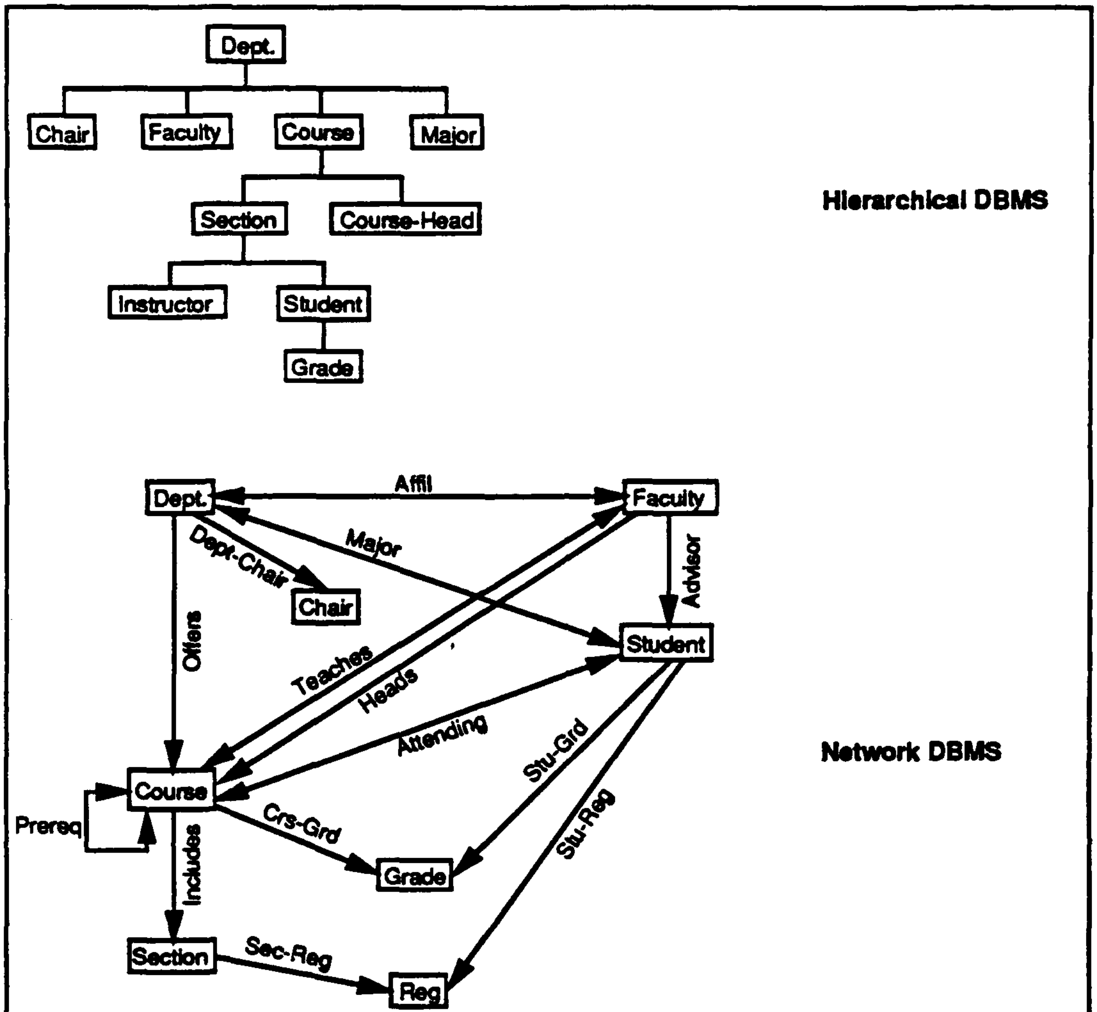
The first DBMSs were developed for business and administrative applications. These record-based DBMSs are usually classified as conventional DBMSs. Although conventional DBMSs work well for business and administrative applications, they do not meet the requirements for engineering applications, and they lack semantic expressiveness. This section identifies and discusses conventional data base models and their deficiencies with respect to the engineering/design environment. The design data base management system (DDBMS) tailored to meet the functional requirements of the design process is also introduced.

A. CONVENTIONAL DATA BASE MANAGEMENT SYSTEMS

A data model defines the overall logical structure of a data base. It provides the structural framework into which the data are placed. The conventional data base models, which dominate the data base systems commonly used today, include the hierarchical, network, and relational models [Refs. 18, 31, 32] (Figure 12). Although the specific modeling constructs of these models vary considerably, each presents the user-level view of a schema in terms of record structures.

1. Hierarchical Data Model

The hierarchical model is tree structured. It is composed of nodes connected by links. The nodes may be grouped into horizontal layers called levels. A hierarchy is a multilevel data model. The tree structure of the hierarchical model implies that each node may be linked to more than one node below itself but to only one node above.



Name	Date of Birth	Job	Work Location
J. Anderson	2-15-1922	congressman	D.C.
J. Carter	10-1-1924	farmer	Georgia
J. Carter	10-1-1924	governor	Georgia
J. Carter	10-1-1924	president	D.C.
R. Reagan	2-6-1911	actor	California
R. Reagan	2-6-1911	governor	California

Relational DBMS

Figure 12. Examples of Conventional Data Base Management Systems

A node represents a type of entity about which information is stored. An entity may be an object such as a wing, a fuselage, or an entire aircraft. Each entity has certain descriptive information associated with it. This information determines the entity type and is referred to as the attributes of the entity. Links represent relationships between the entity types. The links indicate a relationship of one-to-many as they are followed down through the hierarchy.

a. Advantages of Hierarchical Data Model

The major advantage of the hierarchical data model is that it has been successfully used as the basic structure in data base management systems that use the hierarchical data model as the basic structure. The hierarchical data model is also relatively simple and easy to use. Data processing users are very familiar with the hierarchical form. In addition, the hierarchical data model reduces data dependence, and performance prediction is simplified through pre-defined relationships.

b. Disadvantages of Hierarchical Data Model

The disadvantages of the hierarchical data model include difficulties in implementing the many-to-many relationship, which may cause redundancy in stored data. (Although redundancy at the logical level is not necessarily undesirable, since it promotes simplicity, redundancy at the physical level is undesirable.) As a result of the model's strict hierarchical ordering, the operations of insertion and deletion become unduly complex, and hierarchical commands tend to be procedural. Another disadvantage of the model is that deletion of a parent results in the deletion of the children. As a result, users have to be careful when performing a delete operation. Also, child nodes are accessible only through parent nodes because the dominate node type is the root.

2. Network Data Model

The network data model interconnects the entities of an enterprise in a network. In the network data model, the data structures include records and sets. The network data model represents the different types of objects and relationships in the real world. Each type of object is represented as a record type, with the attributes of the object being data fields in the record. A directed arrow connects two or more record types and is used to represent a set type. The record type located at the tail of the arrow functions as the owner record type, and the record type located at the head of the arrow as the member record type. The arrow from owner to member is called a set type. A set type shows a logical one-to-many relationship between an owner and a member.

A network is a directed graph. The network model is a multilevel data model in which each node may be linked to more than one other node in both upward and downward directions--this feature distinguishes the hierarchical model from the network. The network model allows relationships to be established horizontally within levels between different entity types as well as vertically between levels.

a. Advantages of Network Data Model

The major advantage of the network data model is that, like the hierarchical data model, successful data base management systems use the network data model for their basic structures. In addition, the many-to-many relationship, which occurs quite frequently in real life, can be implemented easily. The network data model also provides very good performance and data integrity checking.

b. Disadvantages of Network Data Model

The main disadvantage of the network model is its complexity. The application programmer must be familiar with the logic structure of the data base. The network model also tends to force a single view of data, hence the data are arranged in a rigid, inflexible structure; fixed structural interconnections among data items are not easily molded into a variety of semantic interpretations.

3. Relational Data Model

A relational model is a single-level model consisting of a collection of relations represented in two-dimensional tabular form [Refs. 26, 58]. Associated with the relations is a set of operators that allow for the insertion, deletion, modification, and retrieval of data. A figure for the relational model would simply contain a collection of nodes without any links between them. There are no predefined hierarchies or networks in the relational model. Links needed between nodes are automatically created by the relational DBMS upon demand, and an access path is established to any node.

The rows of a relation are called tuples and its columns are called attributes. All attribute values are drawn from the same domain--they are of the same data type. Each tuple represents an entity and contains a value for each attribute. All tuples are distinct; duplicates are not permitted. Tuples and domains have no order; they may be arbitrarily interchanged without changing the data content and meaning of the relation. Tuples are accessed by means of a key, a single attribute, or a combination of attributes that uniquely identifies a tuple.

a. Advantages of Relational Data Model

The relational data model is easy to understand and use because it is based on the simple concept of a table with rows and columns of data. Users do not face a complicated physical implementation of the model. The relational data model removes the details of storage structure and access strategy from the user interface. The model provides a relatively higher degree of data independence than the hierarchical or network data models. To take advantage of the data independence feature of the relational data model, however, the design of the relations must be complete and accurate.

The relational data model is based on the well-developed mathematical theory of relations. The rigorous method of designing a data base (using normalization[1]) gives this model a solid foundation that does not exist for the other data models. Another advantage of this model is that an unlimited number of relationships can be represented, and thus the extensions that can be made to the set of supportive applications are unlimited. The types of relationships or collections of relationships that can be represented are also unlimited.

b. Disadvantages of Relational Data Model

A major drawback of this model is that the uniformity of structure and the fragmentation into normalized relations compels the user to use queries that are long, repetitive, and tedious, resulting in insufficient performance. Because the relational data model is fundamentally record-oriented, it uses an overly simple data structure to model an application environment. In consequence, the application of a relational model inevitably involves the loss of information and semantics.

4. Model Comparison

While the similarities between the multilevel hierarchical and network models are evident, the network model is more flexible, allowing non-hierarchical or multihierarchical relations to be defined. This added flexibility results in greater representational power, although it still does not afford the representational capabilities of the relational model.

---

[1] Normalization is the process of removing dependencies and redundancies from among the attributes of relations.

Relationships among entities and constraints among attributes impose critical requirements on a data base, and all user queries and updates generally cannot be anticipated prior to the establishment of the data base structure. To satisfy diverse access needs, links may be required between any of the components of the data base; however, the hierarchical and network models have precisely defined links between the nodes. This composition results in fixed data base structures that cannot be easily changed.

The hierarchical model presents difficulties in representing many-to-many relationships. An additional disadvantage, that also occurs in the network model, is that loops are not permitted (relationships cannot be established between a record type and itself). These data models could be particularly limiting when used in an engineering/ design environment, where such relationships are common (for example, ribs connected to adjacent spars or rivets to adjacent panels).

Neither of these disadvantages is found within the relational model. Its use requires knowledge of only one data construct, and its underlying access mechanisms are hidden from the user. The user needs to be concerned with only the content of individual relations. The hierarchical and network models, however, do allow for efficient implementations. Because hierarchy and network links are implemented as pointers, node traversal is direct and fast. A primary disadvantage of the relational model is that it offers less efficient accessing.

An additional advantage of the relational model is its ability to avoid common anomalies through normalization. The concept of normalized relations is an integral part of the relational model, and it promotes the achievement of well-structured data while providing a degree of automatic integrity and consistency checking [Refs. 18, 31, 32].

The results of the data model comparisons are presented in Table 3.

B. DEFICIENCIES OF CONVENTIONAL DATA BASE MODELS

Applying conventional data base models to modeling design and engineering can be problematic. The deficiencies of conventional data base models when used in design and engineering applications are detailed in the following paragraphs [Refs.13, 33-37].

Table 3. Results of Data Model Comparison

FEATURE \ DATA MODEL	HIERARCHICAL	NETWORK	RELATIONAL
Major Applications	Business, Administration	Business, Administration	Business, Administration, Engineering
Data Structure	Records and Sets	Records and Sets	Relations, Domains, and Tuples
Basic Operations	Retrieving and Modifying Data	Retrieving and Modifying Data	Selection, Projection, and Joining
User External View	Static Tree	Directed Graph	Table
Data Independence	Poor	Average	Good
Relationship Binding	Predefined Static Binding	Predefined Static Binding	Dynamic Binding
Semantic Expressiveness	Average	Good	Poor
Data Integrity	Good	Good	Average
Extensibility	Poor	Average	Excellent
Ease of Use	Average	Poor	Excellent
Generality	Poor	Good	Excellent
Cost	Good	Average	Excellent
Performance	Fast	Fast	Slow
Data Shareability	Poor	Average	Poor
Complex Object Representation	Average	Good	Poor
Many-to-Many Relationship	Poor	Average	Average

1. Lack of Semantic Expressiveness

A fundamental problem of the hierarchical, network, and relational data base models is their limited semantic expressiveness. As noted in the preceding paragraphs, the conventional data base models are all fundamentally record oriented. These low-level, record-oriented models use overly simple data structures to model application environments, which results in substantial modeling limitations. The record-based models fail to distinguish the various generic relationships among applications. The relationships, such as *is-inside of*, *is-part-of*, *on-top-of*, commonly used in engineering-oriented applications are difficult to represent with conventional data base models. Consequently, the application of a conventional model inevitably involves the loss of information and only a limited portion of a data base designer's knowledge of the application environment can be expressed.

2. Inability to Handle Engineering Heterogeneous Data Types

Conventional data base models were specifically designed to store and access only formatted alphanumeric data in the form of record files. However, design/engineering data contains a variety of data types--graphical (two-dimensional), geometrical (three-dimensional), mathematical (numerical), procedural, and manufacturing, in addition to alphanumeric. These other data must be extracted with formatted records stored in a data base. In a conventional data base model, such manual processing is arduous and susceptible to errors and delays.

3. Inability to Implement Dynamic Schema

Since schema definition and generation are expensive off-line tasks, conventional data base models only support static schema definition. However, dynamic schema capacities are fundamental for achieving an essential representation of design objects. In conventional data bases, application objects are represented as record structures and are related indirectly through common identifiers, which are character strings that serve as (not necessarily unique) keys to individual records. Thus, in a conventional data base, a subscriber and his claim would typically be associated through an identifier representing a claim number. The subscriber record and the claim record would each contain a copy of the identifier.

Allowing objects to represent themselves instead of using some identifier to represent them makes it possible to directly reference an object from a related one. In record-oriented data base models, it is necessary to explicitly cross-reference between related objects by means of their identifiers; this causes data manipulation to be intricate and semantically confusing. While it is, of course, necessary to eventually represent abstract objects with symbols inside a computer, users (and application programs) should be able to reference and manipulate abstractions as well as symbols; internal representations to facilitate computer processing should be hidden.

4. Limitations of Evolvability

Record-oriented models are also limited in their ability to allow the structure of a data base to support alternative ways of looking at the same information. (The capability of expressing such alternate views may be termed *relativism.*) To accommodate multiple perspectives on the same data and to enable the evolution of new views of existing data, a data base model must support schemas that capture the relationships and similarities between multiple views of the same information.

The primary motivation for relativism is that slightly different views of the same information should be conceptualized as a semantic unit (all of the previous definitions may coexist in the same user view). In conventional models, imposing a single structural organization on the data is generally necessary; this single structure inevitably carries a particular interpretation of the data's meaning. This meaning may not be appropriate for all users of the data base and may eventually become obsolete.

Conventional schemas are also, in a sense, structurally intricate, which affects the evolution of both data base statics and dynamics. Statics expressed in record-based structures are difficult to understand and are therefore so intimately tied to the specifics of the statics that evolutionary changes in the statics are liable to upset the workings of the record-based structures. For example, splitting a relation into two, due to a change in a dependency, is likely to destroy the algebraic operation of any transaction using the original relation. Also, changes in processing requirements, when mapped into record-based languages, often require significant reprogramming efforts.

C. DESIGN DATA BASE MANAGEMENT SYSTEMS

Integrating the engineering design process with computer technology and using sophisticated computer-aided modeling and design systems is an important research area. Data base management technology plays a central role in the integration of CAD/CAM systems. Data bases for CAD systems, so-called design data bases or engineering data bases [Refs. 41, 35-38], have characteristics that differ significantly from those of the business and administrative data bases, which are adequately managed by conventional DBMS. Characteristics of design data bases are identified as follows [Refs. 37-41]:

- *Iterative.* At any stage of the design, incremental changes arise in response to design rule checks, new ideas, or alternative strategies. As a result, iterations for a particular segment of the design process or the entire design must be initiated to reflect these changes.
- *Tentative.* During the design process, all design alternatives must be maintained in the data base, pending final evaluation of the alternatives.
- *Multistaged.* The design is usually separated into various levels or stages, and the designers perform the design on a stage-by-stage basis. The process begins with a product definition, and the design objects are typically designed in several stages, each of which represents a refinement and elaboration of the previous stage. Hence, the earlier stages of design must be available to the designers working on the later stages.
- *Incomplete.* A design data base initially contains only the data determined by initial design decisions. Application programs (design tools) generate more data for future uses, and data continue to be derived until the design is complete. Therefore, the engineering design process is an evolution of a representation. Only when the process is finished is a complete data base achieved.
- *Dynamic.* The design process is a dynamic operation. As the design progresses, design objects and the relationships among them are added, deleted, and modified. In addition, many changes are made based on the results of analysis or the designer's creativity.
- *Extensive Transactions.* A design transaction is defined as a segment of the design process between two states of consistency. In the design environment, reaching a new consistent state is time consuming and may take days or even weeks.

The deficiencies of conventional DBMSs when used in engineering and design applications were quickly recognized. Many engineering organizations and researchers have sought ways to effectively manage data and data bases. The following four approaches for improving the efficiency of data base management systems have been proposed by Ketabchi and Berzins [Ref. 41] (Figure 13).

- Using a special-purpose file manager that views the DBMS as another application tool
- Enhancing the current DBMS by augmenting new capacities
- Building a layer of software and adding it to current DBMSs to compensate for deficiencies.
- Developing a new DBMS, called Design DBMS (DDBMS), equipped with more powerful data models and software facilities required in the design data bases.

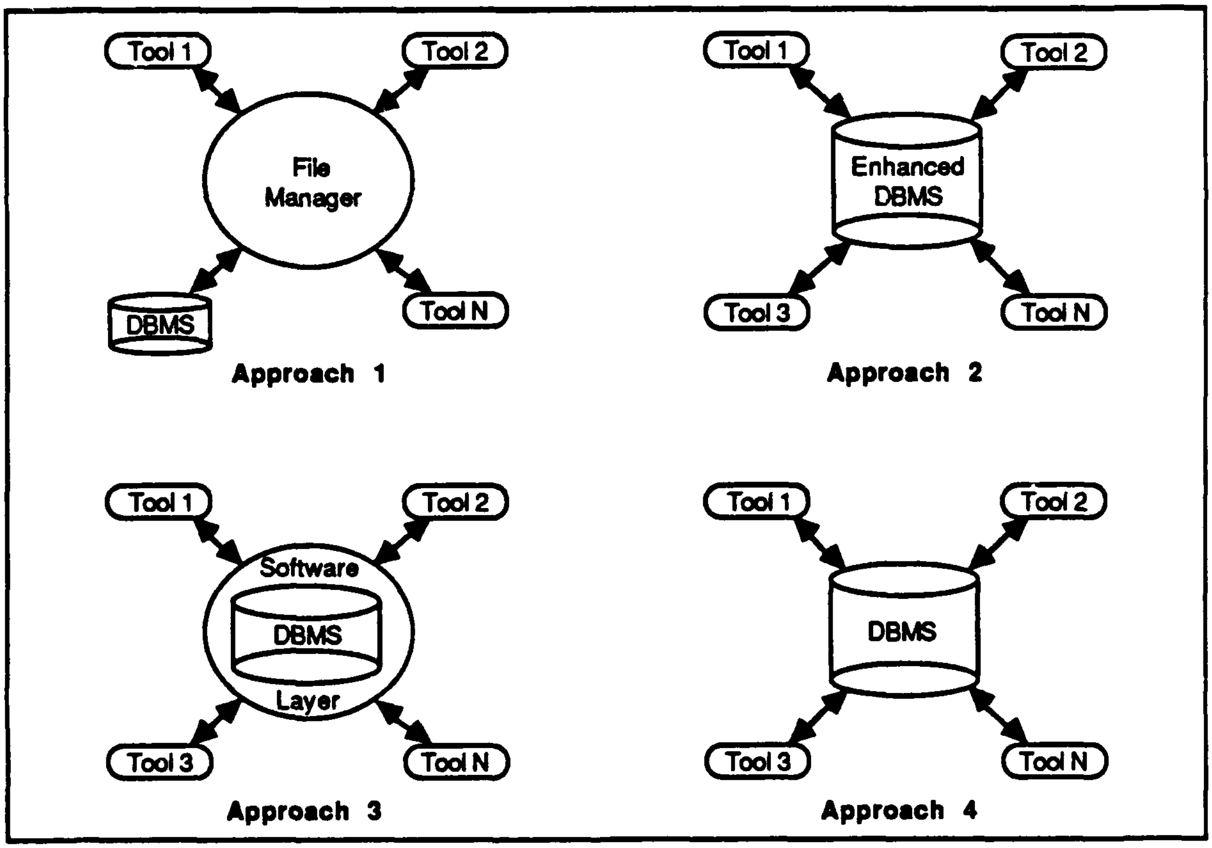


Figure 13. Solutions to Current Data Base Management System Deficiencies

These solutions, however, do have drawbacks. The first approach ignores capacities provided by high-level data models and data base technology. The second approach requires extensive enhancements to the current DBMS and does not provide optimal performance requirements. The third solution does not offer the flexibility and efficiency desired in the CAD environment. The fourth approach, although it requires considerable development effort and advanced software technology, provides designers with a better solution than the three other options.

To develop and implement a DDBMS to manage the engineering and design data, various data/process modeling methodologies have been promoted as a means of providing the informational framework of the DDBMS. Methodologies, such as IDEF, SAMM, NIAM, and semantic data models, together with the improved DBMSs, have been widely used in very large scale integrated (VLSI) circuit design, and mechanical design projects. However, very few have addressed the use of DDBMS in aerospace vehicle design applications.

IV. DESCRIPTION OF DATA/PROCESS MODELING METHODOLOGIES

Because of the deficiencies of the conventional data base model, the three-schema approach to data base design was proposed by ANSI/X3/SPARC Data Base Task Group in 1975 [Refs. 32, 42]. The architecture of this approach includes three levels:

- External schema
- Internal schema
- Conceptual schema.

The external schema supports user views and provides the user interface to the DBMS. One or more external schemas may be provided, each supporting a distinct user view designed for a specific application. The internal schema supports the DBMS and the hardware itself--how the data are physically stored and accessed inside a computer. The conceptual schema serves as an informational model of the enterprise that the data base is to serve and as a control point for additional data base development.

Two advantages are given for the three-schema approach. The first is that the conceptual schema augments the data model with real-world semantics, which are easy to understand and use. The second advantage is the enhancement of data independence, which means that modifying or extending the conceptual schema to capture information need not affect any application program. Based on the three-schema approach, many conceptual data models, such as the Entity-Relationship Model (ERM), the Semantic Association Model (SAM), the Semantic Data Model (SDM) [Ref. 43], the Functional Data Model (FDM) [Ref. 44], and the Object-Oriented Data Model (OODM), have been defined and implemented. They all provide high-level data structuring features to improve the semantic expressiveness of data base conceptual schema and to increase data base accessibility by the end users. In addition to these conceptual data models, a number of modeling methodologies abound to address well-defined information processes. These methodologies, along with the appropriate DBMS, have been applied to different engineering applications. The IDEF methodologies, ERM, SAMM, NIAM, and OODM are described in this section and illustrated with a specific aerospace design example.

A. ILLUSTRATION OF METHODOLOGIES WITH AEROSPACE DESIGN APPLICATION

The test problem adopted in this study covers the early design stages of an aircraft wing composite panel (conceptual and early preliminary design). The composite panel consists of a skin and a certain number of stiffeners, all made with various numbers of ply. With the wide variety of types and fabrics available, many combinations of ply or fabric directions can be used to efficiently sustain the applied load (Figure 14) [Refs. 45, 46]. The ply orientation makes design with composite materials unique because the structure and the material are being designed at the same time. All material properties and strength allowables will vary depending on the ply orientation, and the orientation relies heavily on test data. The stiffness properties alone for a multilayer laminate design can be as many as 21, while metal material design has only 2. In the composite design process, four issues must be addressed:

- Material selection
- Fabrication methods
- Structural integrity
- Environmental effects and protections.

To simplify the test problem, only structural integrity has been considered. Since the elastic modulus (E) of the elements differs because of the various ply orientations, it is necessary to determine E for each element and then determine a transformed area. Once this transformed area is found, the section properties and the average E can be determined and checked for compliance with the allowable compressive stress and limit strain.

Because of the anisotropic properties of the multilayer orientations, considerably more information is needed to perform composite panel design than metal panel design. The following sections describe the various process/data models used for modeling the design of a composite panel subjected to a compressive load. Figures accompany each description to demonstrate the form of the model.

The design of a wing composite panel is considered a well-bounded problem that is sufficiently detailed for purposes of this study, yet not too complicated to implement. The test problem is intended to show how the information (design process and data) are modeled and organized, not to elucidate the completeness and accuracy of the composite panel design. In addition, the test problem is devised to capture and abstract the typical characteristics embodied in the aerospace vehicle and aircraft design processes.

B. IDEF METHODOLOGIES

The IDEF methodology developed by the Air Force's ICAM Program consists of three levels that are used to define functional ($IDEF_0$), informational ($IDEF_1$ or $IDEF_{1X}$), and dynamic ($IDEF_2$) relationships of primarily manufacturing systems (Figure 14). These three levels of communication methodology can be used individually or in combination to provide a comprehensive description of any complex system. This description can then be used to identify the key elements (entities) and relationships, analyze the system evolution, and predict the behavior under certain circumstances.

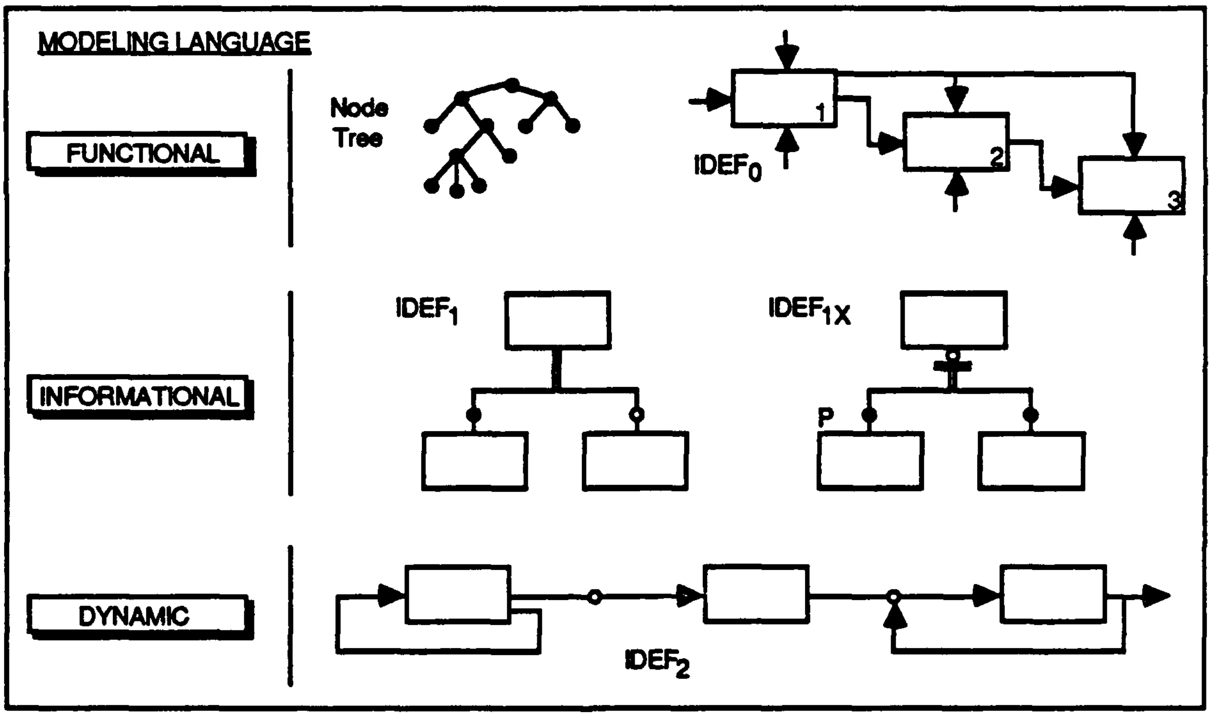


Figure 14. IDEF Methodologies

1. IDEF$_0$ - Function Model

The IDEF$_0$ method employs a diagrammatic technique to hierarchically decompose the entire system into its simplest levels, in terms of their functions in a systematic and logical manner [Refs. 45-49]. The rules that apply to the IDEF$_0$ are illustrated in Figure 15 as follows:

- The box represents an activity that modifies input to produce output.
- Three to six activities are, in general, used for any level of a system.
- Each activity may be subsequently and hierarchically decomposed, refined, and identified.
- Each side of the activity box is used as a location for functional details as follows (see Figure 15):

  Left - inputs to the activity
  Top - controls on the activity
  Right - outputs from the activity
  Bottom - mechanisms required to effect the activity.

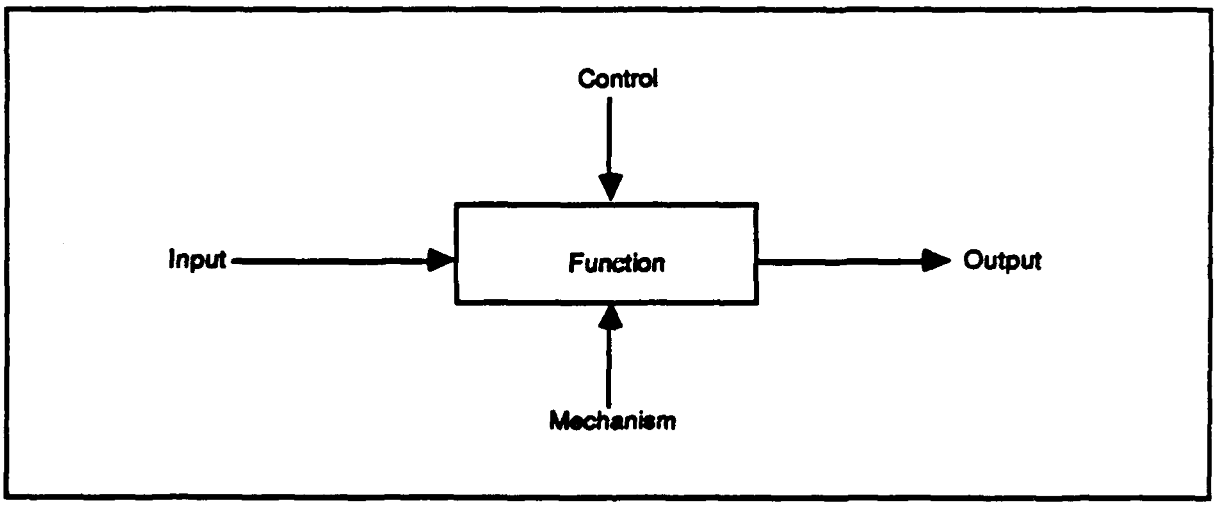


Figure 15. IDEF$_0$ Function Representation

Figure 16 is an IDEF$_0$ representation of the aircraft wing composite panel design application. The connecting lines between activity boxes are identified by nearby labels and may be joined or split to indicate the merging or dividing of information flows. The IDEF$_0$ model is a process model indicating the functional relationships of the various systems, the data flow, and text/glossary sections. Individuals involved in various functions (such as input, output, and controls) are interviewed, and the information resulting from these

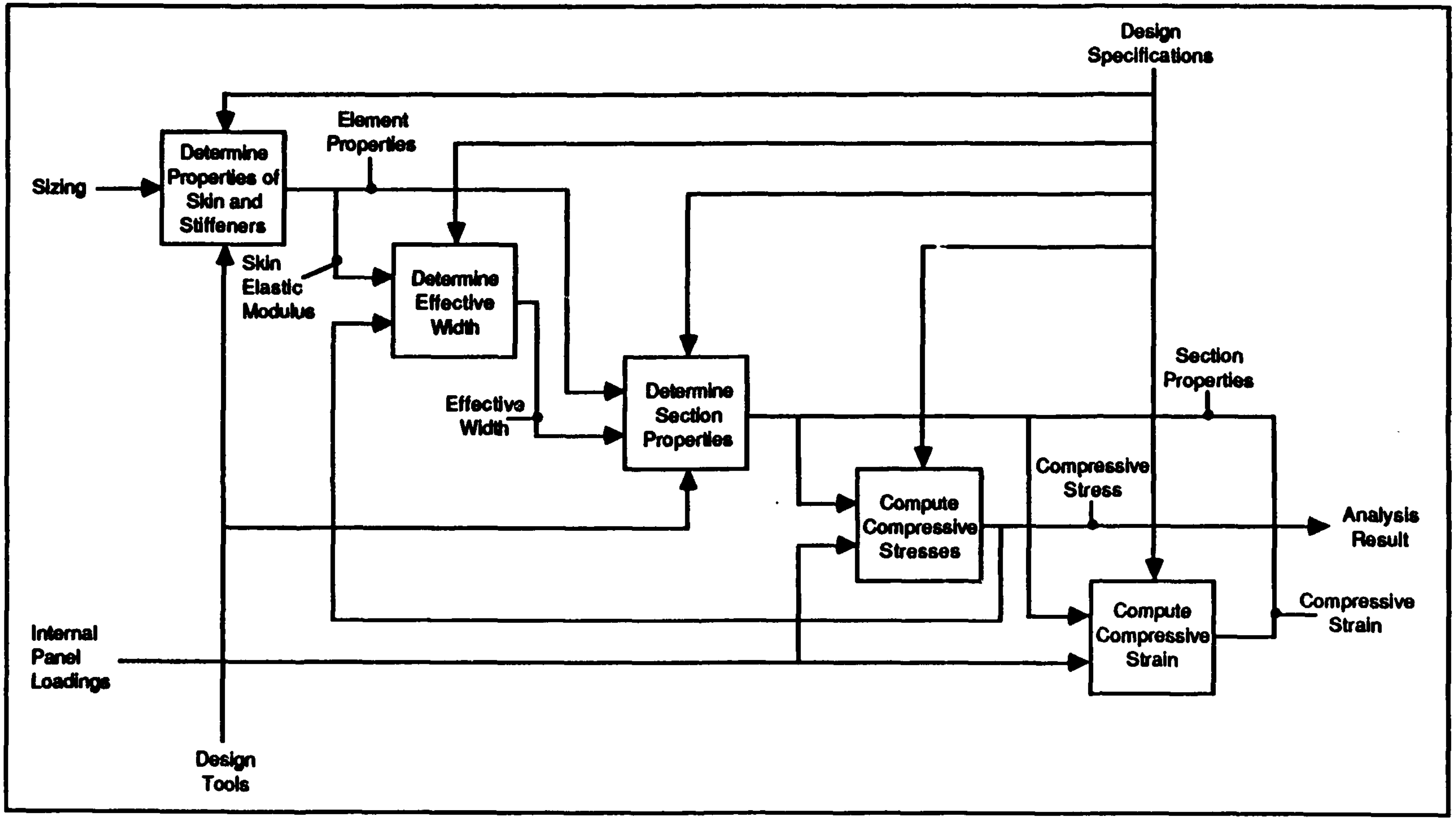


Figure 16. IDEF0 Model for Aircraft Wing Composite Panel Design

interviews is analyzed to define the resources required in each function, while the behavior of the entire system is also considered.

Functions are related through inputs, outputs, mechanisms, and controls. A function will be activated in the systems by receiving inputs to create outputs through guidance of controls, where the activation is performed by mechanisms. $IDEF_0$ can be used to provide a starting point for system improvements or analysis of existing system shortcomings. The $IDEF_0$ methodology enforces a top-down functional modeling approach, an approach that is often found lacking in unsuccessful system designs.

2. $IDEF_1/IDEF_{1X}$ (Extended) - Information Model

$IDEF_1$ is a comprehensive method for describing and analyzing the information of a complex system through a set of rules and procedures for creating information models [Refs. 47, 52-54]. $IDEF_1$ produces graphical diagrams that explicitly represent data semantics in terms of entities (objects), relationships, and attributes (properties).

An entity is an item or an object to which information relates. $IDEF_1$ represents entities by rectangular boxes, and the entity's name is recorded above the box. (Individual entity instances are not represented in the data model.) Characteristics of an entity are known attributes. Each entity instance has a value for each of its attributes. The attribute values are the facts known about the entity instances. Entity attributes are represented in an $IDEF_1$ diagram by names within the entity's box. Relationships, associations between entities, are represented by lines between entity boxes. Each line is labeled with the relationship's name, which is a verb or verb phrase.

With $IDEF_1$, a data model is developed by a top-down analysis of entities and relationships, which is suitable for supporting the full process of developing information systems. $IDEF_1$ is being successfully applied in a variety of enterprises to achieve implementation of the integrated data resources. An $IDEF_{1X}$ representation of the aircraft wing composite panel design application is shown in Figure 17.

3. $IDEF_2$ - Dynamic Model

$IDEF_2$ is a methodology that has been developed to describe the time-varying behavior of manufacturing systems so that computer simulation can be used to generate measures of system performance [Refs. 47, 55, 56].

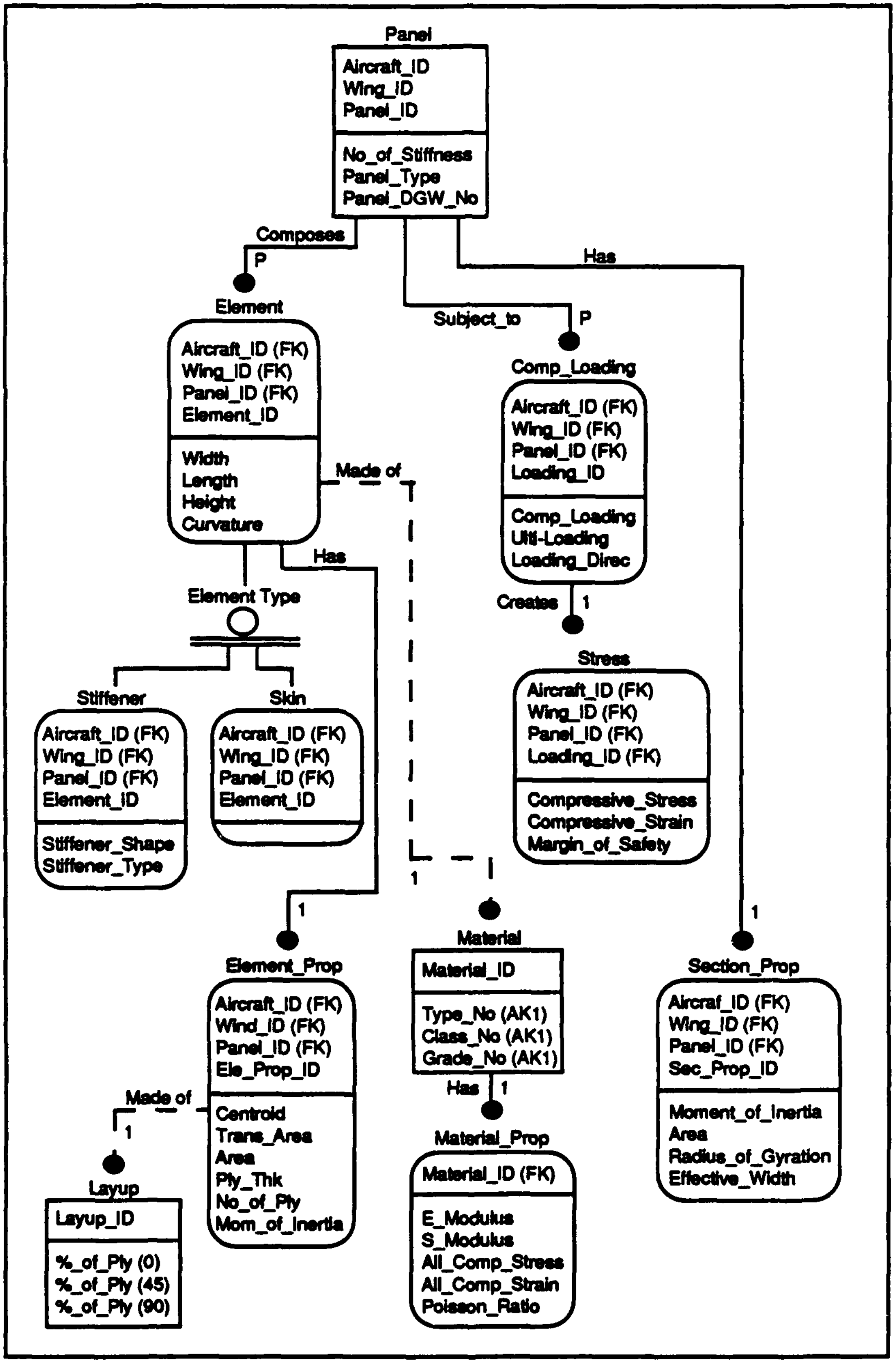


Figure 17. IDEF$_{1X}$ Model for Aircraft Wing Composite Panel Design

To describe a manufacturing system in $IDEF_2$, the system is decomposed into four submodels: Facility Submodel, Entity Flow Submodel, Resource Disposition Submodel, and System Control Submodel. $IDEF_2$ Facility Submodel describes the resources that are used by the system to produce the final products or information. The Entity Flow Submodel details the flow of products or information through facilities. $IDEF_2$ models system behavior by examining the manner in which entities flow through the system and the reaction of the system to the entity flow. The Resource Disposition Submodel is used to describe the disposition of resources when they become available. The Resource Disposition Submodel uses tree structures to organize the actions concerning the resource status of the system. The System Control Submodel describes the occurrences of activities that control but do not prescribe the flow of entities. The System Control Submodel can be used to create entities, alter attributes of entities, and change the capacity of resources.

Each of the submodels within the $IDEF_2$ model contains a graphical component and supporting documentation contained on forms. The graphical components of these submodels have a symbol set designed to facilitate their construction in a straightforward and comprehensive manner. $IDEF_2$ provides a vehicle to predict the dynamic behavior and integrated performance for a large complicated system. Such a system, the aircraft wing composite panel design, is modeled in $IDEF_2$ in Figure 18.

C. SYSTEMATIC ACTIVITY MODELING METHOD

Defined and implemented for the IPAD project, SAMM is a functional model employed to identify the relationships and data flow between each function in a large integrated design system. The SAMM model for the aircraft wing composite panel design application is shown in Figure 19 [Ref. 57].

Similar to $IDEF_0$, SAMM begins with a top-down hierarchical decomposition, which is represented as a tree or node diagram. Each upstream node can branch out into any number of subnodes until the leaves are reached. Each node consists of a set of related activities and can be represented by an activity diagram (AD). The decomposed trace accompanying the AD shows the relationship of each activity to the data flow on the preceding (parent) data model. Each AD contains a series of activities that are connected by data flow arrows. The data flow into the top of an activity represents forward input, and the data flow into the bottom of an activity represents the feedback input. Similarly, the data flow from the right side of an activity represents the forward input, and the data flow

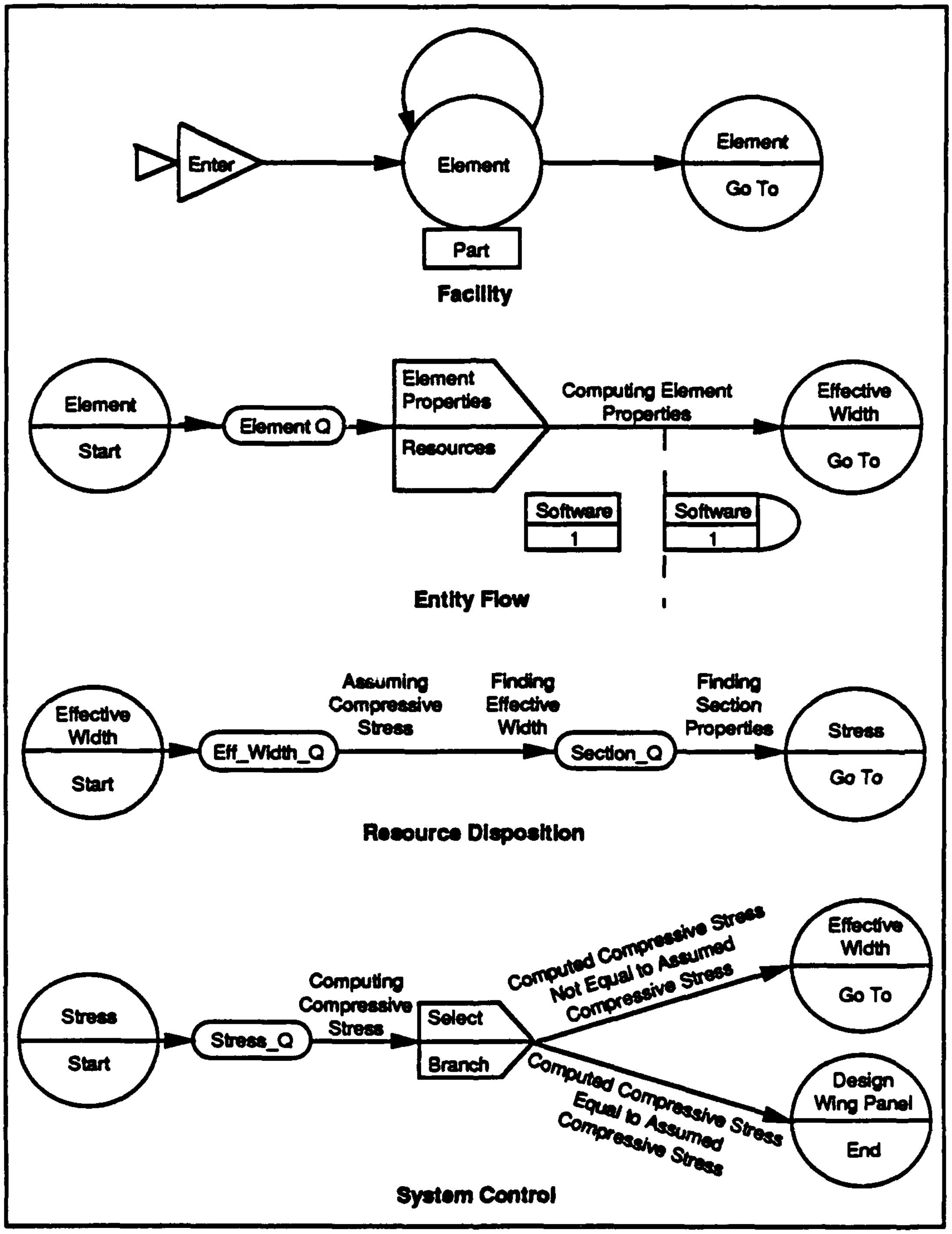


Figure 18. $IDEF_2$ Model for Aircraft Wing Composite Panel Design

PERFORM COMPRESSIVE LOADING ANALYSIS

Data ID	Decomp Trace	Orig/ Dest.	Data Description
1	Ext	–	Design Specifications
2	4	–	Initial Sizing Configuration
3	Ext	–	Internal Panel Loadings
4	–	–	Element Properties for Skin and Stiffeners
5	–	–	Effective Width
6	–	AABBC/ AACAA	Panel Cross-Sectional Properties
7	–	AABBD /AACAB	Compressive Stress
8	–	AABBE /AACAB	Compressive Strain
9	20	–	Resizing Configuration

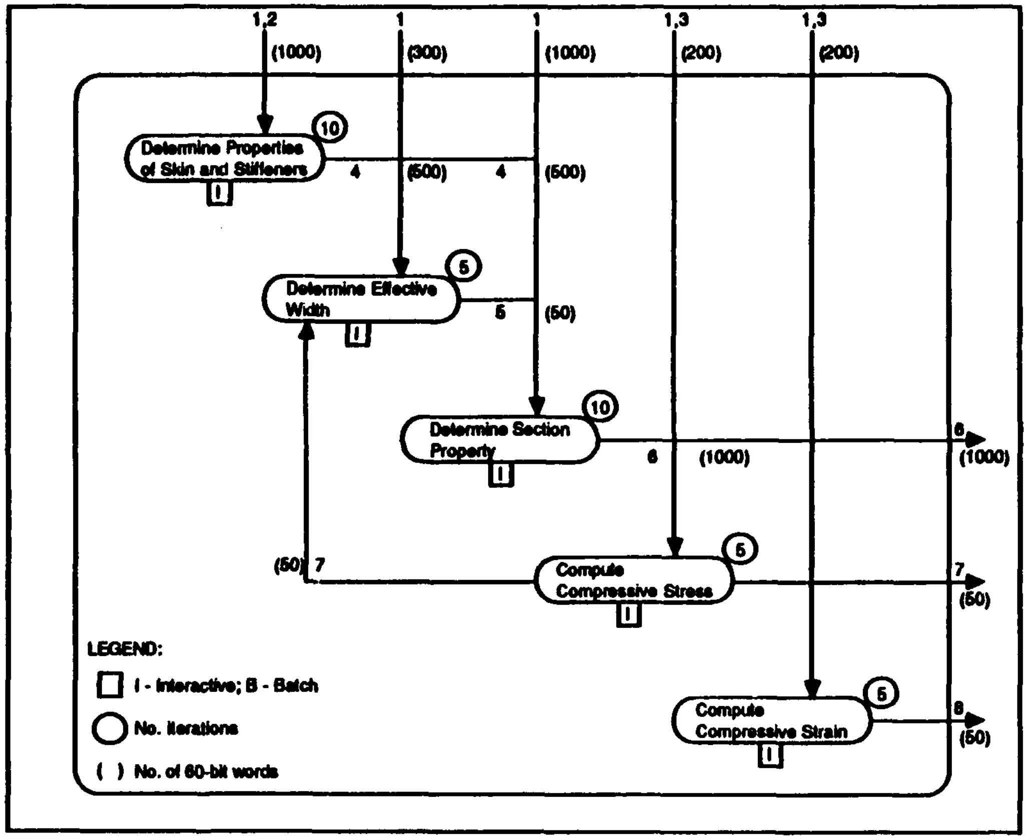


Figure 19. Systematic Activity Modeling Method for Aircraft Wing Composite Panel Design

from the left side represents feedback output. The data flow volume is identified by a number enclosed in parentheses located adjacent to the data identifier number. This number represents the number of 60-bit words transferred on a computer. In addition, the number of iterations of each activity is enclosed in a circle and is estimated for an entire development program cycle. The type of computing support is identified as predominantly interactive or batch.

SAMM is expandable to any level of detail with the hierarchical structure, providing an extremely comprehensive document specifying the activities, their relationships, and functionalities in modeling the design process.

D. NIJSSEN'S INFORMATION ANALYSIS METHOD

NIAM is based on a binary-relationship model using objects and associations (relationships) as two fundamental building blocks to represent the real world [Refs. 58-61]. It provides both information modelers and users with diagrammatic representation, the information structural diagram (ISD).

The objects and associations are represented as circles and edges in the graphical form. Object types are either Nonlexical Object types (NOLOTs) or Lexical Object types (LOTs). Occurrences of NOLOTs cannot be shown; while occurrences of LOTs can be shown. Associations are of two types, bridges and ideas. A bridge type associates a NOLOT and a LOT. An idea type associates two NOLOTs. Further, both bridge and idea types are composed of a pair of roles. The roles describe the nature and semantics of the association between connected objects. With NIAM's ISD, a number of constraints on the associations can also be graphically depicted. The description of constraints allows for devising an algorithm that would pinpoint the state of the lowest possible coupling. With an emphasis on achieving binary semantics associations, coupled with a thorough description of associative constraints, NIAM is able to expose the lowest possible object coupling and functional dependency. This characteristic is commonly known as the third normal form in relational terminology. A NIAM example for the aircraft wing composite panel design application is shown in Figure 20.

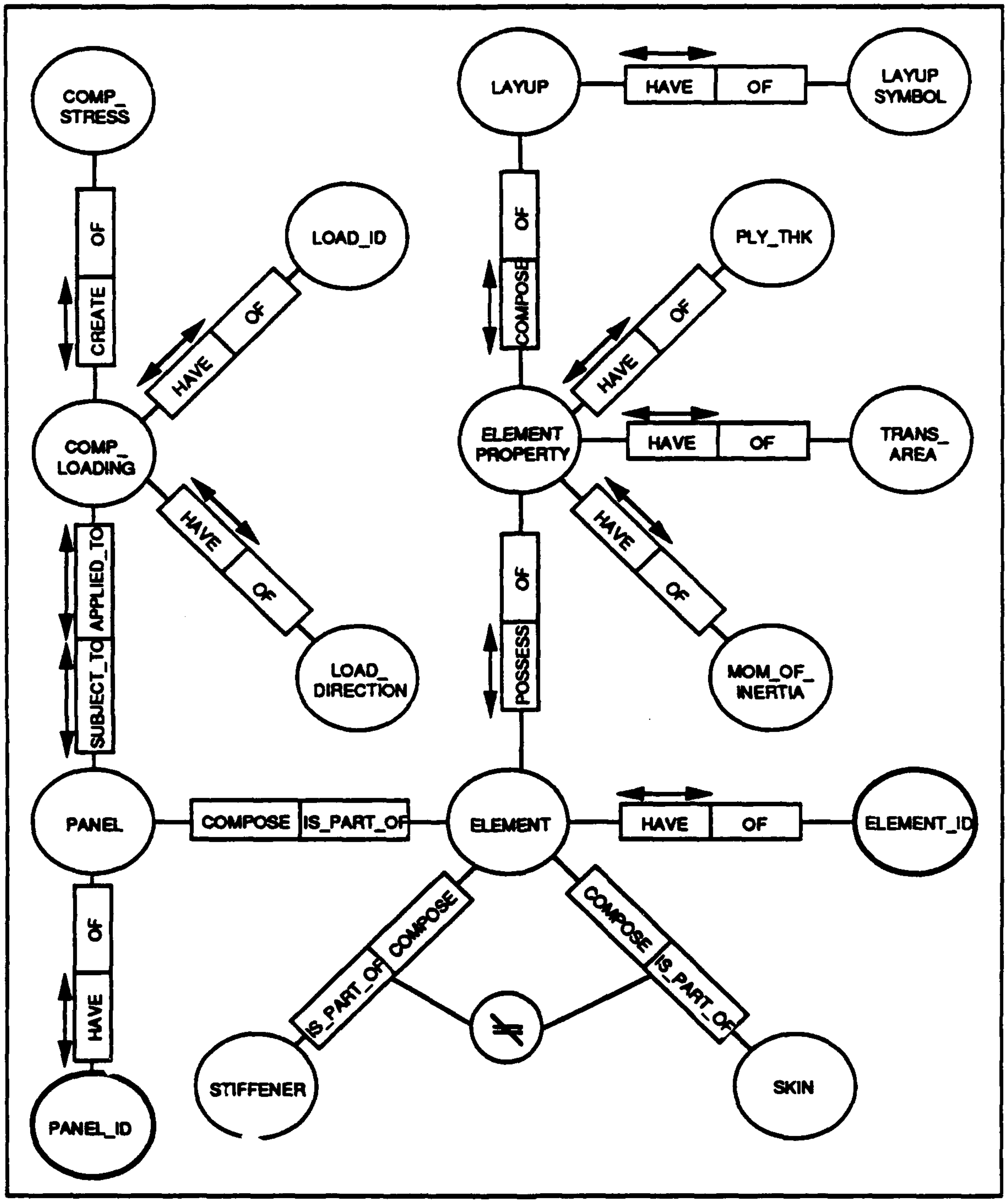


Figure 20. Nijssen's Information Analysis Method Model for Aircraft Wing Composite Panel Design

E. ENTITY-RELATIONSHIP MODEL

ERM was one of the first models to display/represent the correspondence between the semantics of the user's world and the constructs of the relational model [Refs. 62, 63]. ERM contains three primitive conceptual elements--entities, relationships, and attributes (properties)--as representations of the real world. ERM can be implemented using a diagrammatic technique (Entity-Relationship Diagram or ERD) to translate the representation into a logical data base schema in a straightforward manner. The ERD technique represents entities with box symbols and relationships with diamond symbols. In addition, one-to-one, one-to-many, and many-to-many association types are distinguished, and entities and relationships are labeled with meaningful English terms. The ERM for the aircraft wing composite panel design application is shown in Figure 21.

The ERM can adequately, although informally, describe the real world, which is difficult to represent using a conventional data model. One attractive feature of ERD is that it is very easy to map the ERM into the relational data base model with third normal form. In addition, ERM allows the user to have a basic understanding of the underlying logical organization used in the model.

F. OBJECT-ORIENTED DATA MODEL

An OODM [Refs. 64-70], evolved from SMALLTALK-80 (an object-oriented programming language), models all conceptual entities with a single concept--objects. An object may be a primitive object, such as an integer, or a complex assembly of parts, such as an aircraft. The object-oriented data model of the aircraft wing composite panel design application is shown in Figure 22. An object consists of a number of instance variables and/or methods that define the behavior of the object. An instance variable contains the status and the state of that object. The methods are simply procedures that are invoked by messages that are sent by other objects.

A data base may contain a large collection of objects. If every object carries its own instance variable and methods, the amount of information specified and stored can be unmanageably large. In an object-oriented approach, similar objects are grouped into a class, and all objects belonging to the same class have the same instance variables and methods.

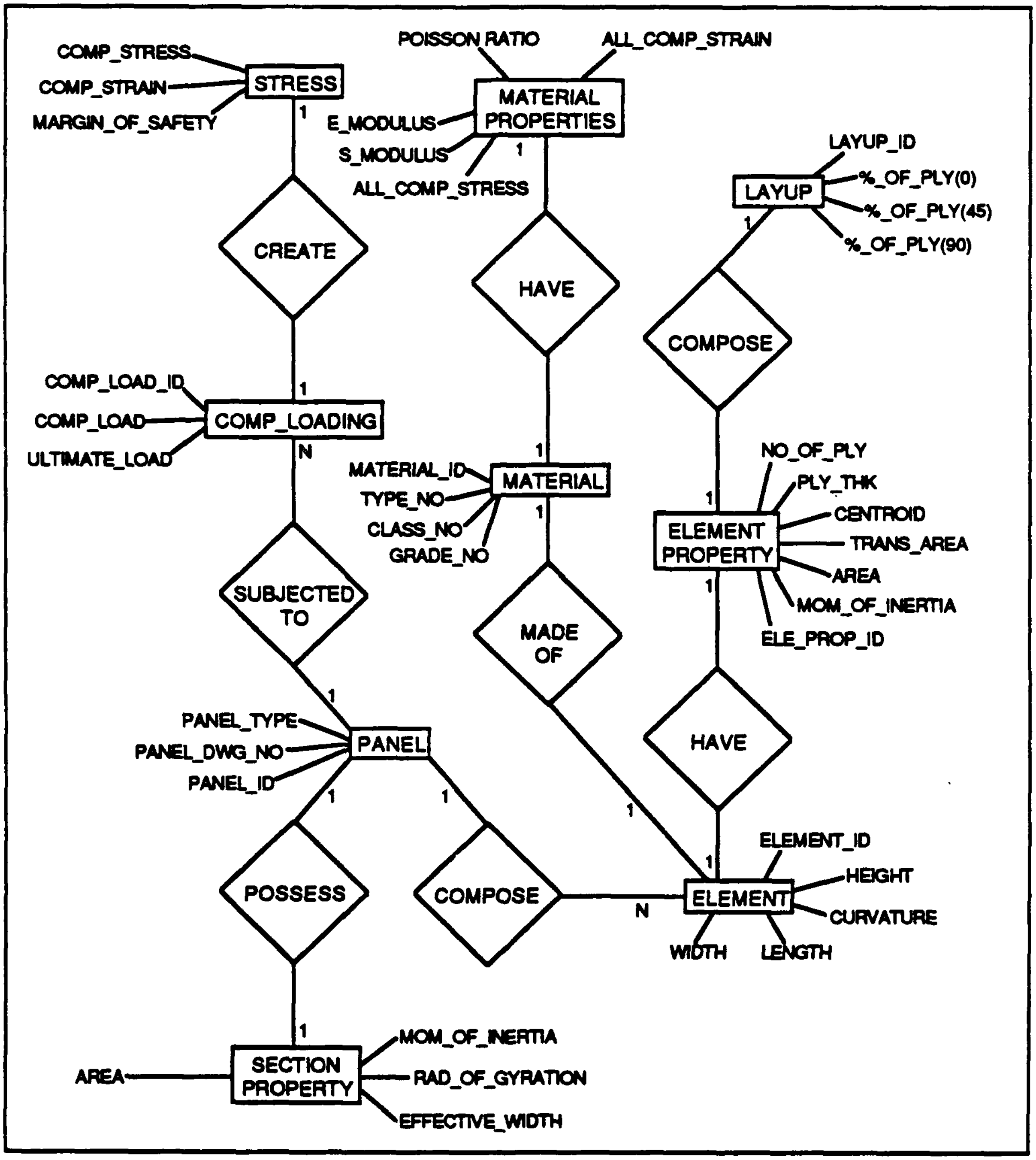


Figure 21. Entity-Relationship Model for Aircraft Wing Composite Panel Design

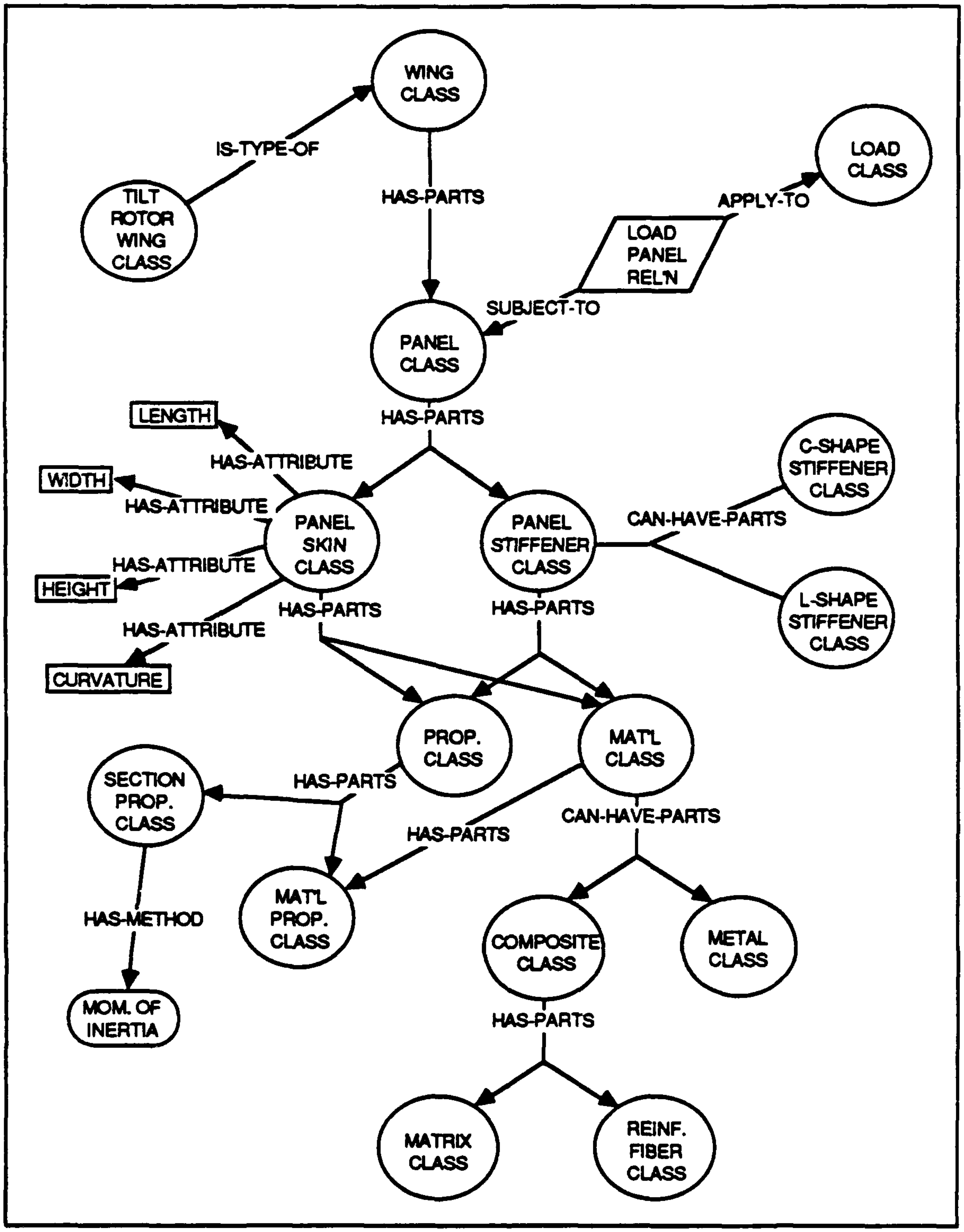


Figure 22. Object-Oriented Data Model for Aircraft Wing Compos... Panel Design

Four characteristics of the OODM are data abstraction, information hiding, inheritance, and dynamic binding. Data abstraction and information binding are important in representing complex engineering design information; dynamic binding is important in developing computer-aided engineering systems that use representations.

Data abstraction is the process of selecting the important properties of a system and representing these properties in a manner understood by the computer. Objects allow very general data abstraction because they represent behavior of an entity independent of the data.

Information hiding is the concept that each object should isolate information from all other objects, with access to the information provided through well-defined interfaces. Objects naturally hide information because the objects include private variables and the procedures (or methods) to operate on the variables. The information presented by the variables and methods can only be accessed by an interface (called a protocol) that is part of the object.

Inheritance allows objects to be organized according to common behavior. For example, an object can be a specialization of a more general object. The specialized object inherits the behavior of the general object and usually adds to or modifies that behavior. In this way, inheritance is used to organize information and simplify the structure of the specialized object using the inherited information.

The final characteristic of the object-oriented data model is the binding of operators with a type of operand at execution.

The advantages of using OODM include

- Ability to manage data from a variety of independent application programs under the same interface (such as adding or changing programs)
- Ability to improve the representation of information, such as spatial data, which are handled with conventional data models
- Ability to manage part hierarchies and recursive data, which are not supported by conventional data models
- Ability to develop software programs, such as modelers, without being concerned with data base management techniques
- Ability to support different views of the same application, which is essential in integrating and managing design data bases.

V. EVALUATION OF DATA/PROCESS MODELING TECHNOLOGIES

While much has been written about applying the modeling methodologies described in the preceding section to design processes, such as VLSI design, literature focusing on the aerospace engineering application is limited. This chapter contains an evaluation of these methodologies for use in modeling aerospace vehicle design.

The functional requirements needed to provide designers an optimal design environment in which to carry out the aerospace vehicle design process are first identified. These functional requirements or specifications serve as the skeleton for development of an ideal methodology. The test problem is then used to evaluate these methodologies by pinpointing their features, assets, and liabilities and rating their capacities against the identified functional requirements.

A. FUNCTIONAL REQUIREMENTS

The functional requirements [Refs. 20, 38, 68, 73] for an ideal process/data modeling methodology embody features that provide designers an optimal design environment, by addressing the special features of the complex aerospace vehicle design process (described in Section II-A). These features are

- *Dynamic Schema.* A data schema defines the framework or representation of the design object and the associated relationships. Since the design process is highly iterative and dynamic, the data model must support the evolving nature of the data schema--the schema must be flexible enough to support modification and extension.
- *Versions, Alternatives, and Revisions.* Throughout the design process, artifacts can often have several versions, alternatives, and revisions, which have different descriptions and definitions. The process/data model must provide a mechanism for storing and managing multiple versions, alternatives, and revisions so that designers can retrieve information about early designs, test new design ideas, and compare design options.

- *Complex Object Representation.* The design object is usually an assembly of many part objects that may contain subpart objects or may contain lower-level part objects. The design data base model should support the hierarchical relationship between object types and the inheritance of certain characteristics and properties.
- *Design Transactions.* The various transactions of different designers must be coordinated so that the entire system operates smoothly. In conventional data base models, a transaction has an all-or-nothing interpretation and is not suitable to some of the design transactions that require extensive transaction times. The design data base model must be able to handle the latter kind of transaction as well.
- *Multiple Views.* Different designers are interested in different portions of the system. Consequently, different views of the entire system must be available for different designers/teams. (Informally, a view can be thought of as a portion of the window into a portion of the data base.) The model should be flexible enough to allow the dynamic definition and movement of a view, instead of forcing the predefinition of possible views.
- *Heterogeneous Data Types.* The design process involves the use of many data types, such as graphical data, textual data, procedural data, mathematical data, and manufacturing data, which differ greatly in their representations and structures. The process/data model should support all data types.
- *Data Independence.* The ability of application programs to have a constant logical view of the data base structure, independent of the data base's realization on a physical storage medium is termed *physical data independence.* Data models should exhibit this property as well.

B. EVALUATION MATRICES

To evaluate the effectiveness of the methodologies in aerospace engineering applications, each methodology was used to model the wing composite panel subjected to a compression load. The features, assets, and liabilities identified through the evaluation are listed in Tables 4-10.

The comparison of the evaluations for the process/data modeling methodologies is shown in the evaluation matrix (Table 11). The first column lists all of the characteristics and functional requirements (identified in Section II-A and Section IV-A, respectively). Each methodology is rated in terms of its capability to model these characteristics and requirements. These subjective ratings are based on how the methodologies performed in

Table 4. Summary of $IDEF_0$ Methodology

ITEM \ MODEL	$IDEF_0$
Features	• It is a functional model that describes a complex system and interrelated information/ object transfer. • It provides graphics, texts, and forms that permit the system designers to quantify the existing system, propose system enhancements, and evaluate their effects in a logical way. • It strongly reinforces the top-down functional modeling approach. It gradually introduces greater levels of detail through the diagram structure of the model.
Assets	• It permits an individual to work on different aspects of the total system design and yet be consistent in terms of final systems integration. • It permits complete system encapsulization in a standardized, documented form. • It permits the user to specify a complete system design to the desired level of detail. • It is a clear, concise specification methodology currently available to functionally describe total system design.
Liabilities	• Development time is too lengthy. • It is quite complex and time consuming to read. • It only has a static representation of facility. It is not able to define the system in terms of dynamic representation. • The function names between two different modelers can be inconsistent due to their different views about the system. • Sometimes it has difficulty in pinpointing a problem area within the system.

Table 5. Summary of $IDEF_1/IDEF_{1X}$ Methodology

MODEL / ITEM	$IDEF_1/IDEF_{1X}$
Features	• It comprises three primary elements: – Entities (classes of things of information) – Attributes (classes of kinds of information) – Classes of relations between entities. • It incorporates the necessary graphics, texts, and forms to inject an organized discipline into the process. • It provides for the measurement and control of the progressive development of the model through the routine of the modeling discipline. • It is a coherent language that supports the development of conceptual schemas.
Assets	• It produces graphical diagrams that explicitly represent data semantics. • It represents a broad range of detail, making it suitable for supporting the complete process of developing information systems. • It is independent of any DBMS and application tools. • It has been successfully applied in a variety of enterprises to achieve implementation of integrated data resources. • It provides a modularity that can protect against inaccuracy, incompleteness, inconsistency, and imprecision. • It supports disciplined, coordinated teamwork.
Liabilities	• It describes only the static behavior of information in a system. • Considerable knowledge is required for implementation, and building the data model is time consuming. • Inexperienced users often generate a non-normalized form and later cause data base anomalies.

Table 6. Summary of $IDEF_2$ Methodology

MODEL / ITEM	$IDEF_2$
Features	• It describes a time-varying behavior in a systematic way such that the descriptions can be analyzed using computer simulations to generate a measure of system performance. • It decomposes into four submodels (graphic components): --Entity flow networks --Resource disposition trees --System control networks --Facility diagrams. • It models system behavior by examining the manner in which entities flow through the system and the reaction of the system to the entity flow.
Assets	• It is suitable for measuring the performance in terms of time. • It can model probability of occurrence, personnel involvement, decision making, and interactions among activities and events. • It is suitable for modeling the dynamic behavior of bounded systems, such as manufacturing processes. • It predicts and experiments with a system's dynamic behavior without implementing and building the system. • It makes use of computer simulation techniques and reduces human error.
Liabilities	• It is difficult to understand and implement due to complexity. • It can handle only well-bounded manufacturing processes. It is not suitable to model an unbounded system, such as a design activity.

Table 7. Summary of Systematic Activity Modeling Method

MODEL / ITEM	SYSTEMATIC ACTIVITY MODELING METHOD (SAMM)
Features	• It provides a systematic approach by using a top-down hierarchical decomposition technique approach. • An activity diagram (AD) is used to show the interrelationships between activities by indicating data and data flow through their relationships. • It can be used to model the design networks that are the fundamental building blocks for the design process. • It is designed to be expandable to the level of detail desired by the designers.
Assets	• It allows the individual to construct the model in a parallel and modulized manner without involving the details of other activities. • It provides information such as the number of iterations, the quantity of data, and whether the activity can be performed using computers. • It permits the designer to specify a complete system design to the desired level of detail. • It permits the design to be reviewed and examined by many individuals, and comments by these individuals can be incorporated in a consistent, standardized manner. • The cost and time drivers can be quantified.
Liabilities	• It does not indicate a specific sequence or flow as evolving over time. This fact is frequently misunderstood by users. • It does not have information about the involvement of mechanisms such as design tools, computer hardware or personnel. • It encourages the designers to concentrate on individual activity without seeing the process as part of the entire system. • It is a static representation of the activity, which may be problematic since designers have difficulty perceiving the design process in terms of static data flow.

Table 8. Summary of Nijssen's Information Analysis Method

MODEL / ITEM	NIJSSEN'S INFORMATION ANALYSIS METHOD (NIAM)
Features	• It is a binary-relationship conceptual data model. • It is a means of capturing information requirements in user-understandable terms, modeling and analyzing the requirements in a formal information model, and translating conceptual information requirements into implementable specifications. • Relationships between object types are derived through entity-joins rather than symbol-joins. • It is a rule-based modeling technique that can be easily mapped into the data base schema and data specifications up to the third normalized form using functional decomposition and an information structural diagram (ISD).
Assets	• It is easy for non-technical people to use because schemata defined in terms of the model can be read almost like a natural language. • It supports a variety of constraints that are not available in the conventional data models. • Users have complete freedom to override the form suggested by NIAM and dilute the high level of normality. • It uses a semantic binary association between objects in generalized object classes; therefore, it is capable of modeling any environment. • Information can be easily automated by the computer algorithms to transform the conceptual schema into a logical data base schema.
Liabilities	• It is not considered a "real" data model since it does not provide a well-defined set of data manipulation operations. • It does not provide capacities for view definition. • It is inefficient, even with simple queries, requiring a greater number of joint operations than conventional data models.

Table 9. Summary of Entity-Relationship Model

MODEL / ITEM	ENTITY-RELATIONSHIP MODEL (ERM)
Features	• It is one of the earliest conceptual data models. • It supports the top-down approach. • It consists of three basic constructs: entities, relationships, and attributes. • It can model composite entities or their relationships. • The Entity-Relationship Diagram (ERD) provides users a visual immediacy that makes ERM a popular conceptual data model.
Assets	• The ERM's basic construct is very simple to represent and learn. • The ERD is a comprehensive and simple diagrammatic technique. • Many-to-many relationships are easy to implement. • ERD can be easily mapped into a relational data base structure with up to the third normal form. • It is supported by the well-developed entity-relationship modeling tools.
Liabilities	• It assumes that an entity can be represented by a single relation. • Even if classified as a semantic data model, ERM still cannot provide sufficient semantics for engineering design objects. • It provides the modelers with a great deal of freedom to model the enterprise; hence, the models generated by different individuals can have many discrepancies.

Table 10. Summary of Object-Oriented Data Model

MODEL / ITEM	OBJECT-ORIENTED DATA MODEL (OODM)
Features	• It models all of the conceptual entities with a single concept--objects. • Each object encapsulates data and procedures to operate on the data. • It has four characteristics: data abstraction, information hiding, inheritance, and dynamic binding. • It provides a hierarchy of types of objects and the ability to inherit the properties of the parent object types. • It allows application programs to view a class of abstract data objects completely in terms of a set of characterizing operations.
Assets	• Complex design entities can be represented more directly, with less encoding, meaning fewer levels of indirection. • It offers fast response in design applications. • Update operations and constraints are an integral part of the data base. • Data independency is maintained. • An efficient programming language interface can be developed. • Iterative and tentative nature of design is supported. • Multistage nature of design is supported. • Dynamic schema and data base operations are extendable. • Data can be shareable. • Versions, alternatives, and revisions can be easily implemented.
Liabilities	• The concept is difficult to implement. • The dynamic binding mechanism has high run-time costs. • A variety of the object-oriented paradigms, each defining different terminologies and meanings, cause inconsistencies and confusion to designers not proficient in DBMS.

Table 11. Comparison of the Methodologies

MODEL / FEATURE	$IDEF_0$/ $IDEF_1X$	$IDEF_1$	$IDEF_2$	SAMM	NIAM	ERM	OODM
Version Control	Poor	Poor	Poor	Poor	Average	Average	Good
Iterative and Tentative Process	Average	Poor	Average	Average	Poor	Poor	Good
Dynamic Schema	Poor	Poor	Average	Poor	Poor	Poor	Average
Design Transaction	Average	Poor	Good	Average	Poor	Poor	Poor
Modeling Style	Top-Down	Top-Down	Front-End	Top-Down	Bottom-Up	Top-Down	Top-Down
Multiple View	Poor	Poor	Poor	Poor	Poor	Poor	Average
Ease of Use	Good	Average	Poor	Good	Excellent	Excellent	Poor
Integrity Checking	Poor	Average	Good	Poor	Excellent	Good	Good
Local Constraint	Poor	Poor	Poor	Poor	Poor	Poor	Poor
Software Support	Good	Good	Poor	Good	Excellent	Excellent	Poor
User-Defined Relationship	Poor	Average	Poor	Poor	Excellent	Good	Excellent
Complex Object Modeling	Poor	Average	Poor	Poor	Average	Good	Excellent
Data Shareability	Good	Average	Poor	Good	Good	Good	Excellent
Supports Multidiscipline Team Work	Excellent	Good	Poor	Excellent	Poor	Poor	Average

the test problem. The results indicate that none of the methodologies meets all of the specified functional requirements and characteristics of the design process, although the object-oriented data model was found to be the best data modeling method for modeling the aerospace vehicle design process. Combining two methodologies (one data model and one process model) seems to be the best strategy for covering the requirements of modeling the aerospace vehicle design process; however, the effort and time required to do so may be prohibitive.

C. ADDITIONAL EVALUATIONS

In addition to identifying the predominant features, assets, and liabilities of these methodologies, which is essential to fully evaluate them, other intangible features, such as the assumptions the methodologies are based on and the skills required for use, are also important. The application of these methodologies requires not only a knowledge of the steps and techniques involved, but also a comprehension of the underlying concepts, philosophy, and scope.

1. Assumptions

All methodologies are based on certain assumptions, although they are not always explicitly stated in the documentation of the model. Most methods are based on the common assumption that information can be modeled and that the models used in each methodology are adequate for this purpose. More importantly, the mapping of these models to implementation or physical models is assumed to be a simple task. NIAM uses object types and associations; IDEF_{1X} uses entities, attributes, and relationships; ERM uses entities and relationships; and OODM uses only objects as the basic constructs.

Other important assumptions are concerned with the decomposition of the systems and the sequencing of the tasks. SAMM, IDEF, and ERM are based on the assumption that a system can be hierarchically decomposed and partitioned using a top-down approach. NIAM is based on the assumption that the integration of information systems in a bottom-up fashion can be applied to produce logical data models.

2. Skills Required for Use

The skills needed by the designers to use these methodologies differ substantially. For some, no particular data processing skills are necessary; NIAM and ERM have been successfully used by novices, while others, such as OODM and IDEF, require extensive experience. These methodologies can be used individually or in combination to provide a comprehensive description of any complex system, such as modeling of the aerospace vehicle design process. Although the skill levels necessary to use the methodologies vary significantly, the complete knowledge and understanding of the system modeled is required.

3. Scope

The scope of the methodologies also varies among the application areas. The maximum benefits of each methodology can be obtained only by using the method in the specific application area for which it was developed. For example, IDEF methodologies were established for modeling the function, information, and dynamics of manufacturing systems. SAMM is best suited for modeling design activities in the system development phases. ERM and NIAM were designed to be the modeling methods for the entire scope of an information system. OODM, however, is suitable for modeling engineering design objects.

4. Graphical Representation and Software Support

All methodologies support graphical representation and techniques to facilitate and simplify the modeling process. $IDEF_0$ and SAMM use rectangular boxes to represent activities and arrow lines to relate them. The ERD technique employs boxes (entities) and diamonds (relationships). $IDEF_1$ and $IDEF_{1X}$ apply rectangles and lines to represent the entities and relationships. OODM and $IDEF_2$ have many graphical constructs with different representations that can be difficult for novice designers to grasp. Almost all of the modeling methods can be automated and aided by commercially available software. IDEF methods are supported by ECLIPSE developed by DACOM. NIAM is supported by IAST and RIDL, developed by CDC. OODM is supported by VBASE developed by Ontologic.

VI. CONCLUSIONS AND RECOMMENDATIONS

The aerospace vehicle design process is a complex activity that often requires the effort of many individuals over an extensive period of time. The entire scope of the design process is driven by design information created during the design operations. To effectively manage the design information, various CAD/CAM and data base technologies and techniques, such as integrated data base concepts, distributed data base concepts, and semantic data modeling methods, have been developed and implemented. One of the key issues for ensuring the effective management of engineering information is the use of a DBMS specifically tailored to engineering applications. Conventional DBMSs, developed for business- and administration-oriented environments, cannot fulfill the functional requirements for engineering applications. In light of the deficiencies of conventional DBMSs, many data/process modeling methodologies have been advocated and implemented. Such methodologies were developed to serve the needs of particular engineering tasks and previously were not sufficiently evaluated in terms of their relevance to aerospace vehicle design. The IDA study covered seven data/process modeling methodologies ($IDEF_0$, $IDEF_{1X}$, $IDEF_2$, NIAM, SAMM, ERM, and OODM), which were evaluated by testing their application to aircraft wing composite panel design. The results of this evaluation are summarized in this paper.

The study indicated that none of the existing modeling methodologies is adequate for supporting the overall aerospace vehicle design process. The OODM seems to possess many of the features required for the ideal design decision support system for modeling the aerospace vehicle design process. Some of the features that the OODM lacks are embodied in other methodologies. It is felt that research toward an extended information modeling methodology, formed by combining the OODM data model with a process model (such as $IDEF_0$ or SAMM), may provide the optimal design decision support environment. Such a modeling method must be developed, implemented, and tested to provide critically needed support of future information-driven aerospace design processes. Large-scale test bed problems, such as the XV-15 tilt-rotor composite aircraft wing structure or avionics control systems, should be used to evaluate the information methodologies assessed in this report, as well as any future methodology developments.

REFERENCES

1. Fulton, R.E., "A Framework for Innovation," *Computers in Mechanical Engineering*, March 1987, pp. 26-40.

2. "Decision Support Requirements in a Unified Life Cycle Engineering (ULCE) Environment, Volume I: An Evaluation of Potential Research Directions," ULCE DSS Working Group, IDA Paper P-2064, May 1988.

3. Brei, ML, W.E. Cralley, D. Dierolf, D.J. Owen, K.J. Richter, and E. Rogan, "Architecture and Integration Requirements for an ULCE Design Environment," IDA Paper P-2063, April 1988.

4. Alsina, J., J.P. Fielding, and A. J. Morris, "Progress Towards an Aircraft Design System," *Computer Applications in Aircraft Design and Operations*, T. K. Smurthy, Ed., 1987, pp. 229-247.

5. Kinoglu, F., D. Riley, and M. Donath, "Expert System Model of Design Process," *IEEE/ACM 1983 Design Automation Conference*, 1983, pp. 7-14 to 7-27.

6. Meyer, D.D., "Development of Integrated Programs for Aerospace-Vehicle Design (IPAD) - Reference Design Process," NASA Contract NASA-14700, NASA Contract Report CR-2981, Boeing Commercial Airplane Company, March 1979.

7. Heldenfels, R.R., "Integrated Computer-Aided Design of Aircraft," *NATO AGARD Conference Proceedings in Aircraft Design Integration and Optimization*, AGARD-CP-147, October 1973, pp. 16-1 to 16-12.

8. Barnes, R., "Technology for the Product & Process Data Base," *Proceedings of a National Symposium for IPAD II--Advances in Distributed Data Base Management for CAD/CAM*, Denver, CO, April 17-19, 1984, NASA Conference Publication No. 2301, NASA, Washington, DC, April 1984, pp. 35-41.

9. McKenna, E.G., "Design Verse Manufacturing Data Base Management Requirements," *Proceedings of National Symposium for IPAD II--Advances in Distributed Data Base Management for CAD/CAM*, Denver, CO, April 17-19, 1984, NASA Conference Publication No. 2301, NASA, Washington, DC, April 1984, pp. 51-79.

10. Dittrich, K.R., A.M. Kotz, and J.A. Mulle, "A Multilevel Approach to Design Database Systems and its Basic Mechanisms," *Proceedings of IEEE 1985 Compint - Computer Aided Technologies*, 1985, pp. 313-320.

11. Peterson, J.G., G. Nadler, and M. Chignell, "Perspectives on Design, Problems, Processes and Purposes," *Proceedings of the 1987 International Conference on Machines, Man, and Cybernetics*, 1987, pp. 980-985.

12. Madison, D.E., T.J. Wilbur, Jr., and C.T. Wu, "Data-Driven CIM," *Computers in Mechanical Engineering*, March/April 1987, pp. 38-45.

13. Ebel, J.L., and A.J. Gadient, "The VHSIC Engineering Information System Program," *Proceedings of IEEE 1986 National Aerospace and Electronics Conference, NAECON '86*, Vol. I, May 1986.

14. Ralston, D., and T. Munton, "Computer Integrated Manufacturing," *Computer-Aided Engineering Journal*, August 1987, pp. 167-174.

15. Hsu, C., and C. Skevington, "Integration of Data and Knowledge in Manufacturing Enterprises: A Conceptual Framework," *Journal of Manufacturing Systems*, Vol. 6, No. 4, pp. 421-428 and 277-285.

16. Dittrich, K.R., A.M. Kotz, and J.A. Mulle, "A Multilevel Approach to Design Database Systems and its Basic Mechanisms," *Proceedings of IEEE 1985 Compint - Computer Aided Technologies*, 1985, pp. 313-320.

17. Bray, O.H., "Computer-Integrated Manufacturing: The Data Management Strategy," Digital Press, Bedford, MA, 1988.

18. Atre, S., "Data Base: Structured Techniques for Design, Performance, and Management," published by John Wiley & Sons, Inc., 1980.

19. Hatchell, B.K., G. Benatar, R.E. Fulton, and C.L. Blackburn, "An Aid for Instruction on Integrated Engineering Design and Support," *Managing Engineering Data: The Competitive Edge*, The 1987 ASME International Computers in Engineering Conference and Exhibition, New York, NY, August 1987, pp. 87-95.

20. Staley, S.M., and D.C. Anderson, "Functional Specification for CAD Databases," *Computer-Aided Design Journal*, Vol. 18, No. 3, April 1986.

21. "IPAD II--Advances in Distributed Database Management for CAD/CAM," *Proceedings of a National Symposium for IPAD II--Advances in Distributed Data Base Management for CAD/CAM*, Denver, CO., April 17-19, 1984, NASA Conference Publication No. 2301, NASA, Washington, DC, 1984.

22. Garrocq, C.A., J.J. Hosek, et al., "Feasibility Study of an Integrated Program for Aerospace-Vehicle Design (IPAD) System," NASA Contract NAS1-11431, Contract Report CR-132403, General Dynamics Corporation, August 1973.

23. Southall, J.W., "Development of Integrated Programs for Aerospace-Vehicle Design (IPAD) - Information Processing Requirements," NASA Contract NAS1-14700, NASA Contract Report CR-2981, Boeing Commercial Airplane Company, June 1977.

24. Fulton, R.E., "Overview of Integrated Programs of Aerospace Vehicle Design (IPAD)," NASA Technical Memorandum TM-81874, 1980.

25. Diggins, R.M., "Preliminary Design of a Future Integrated Design System," *Proceedings of National Symposium for IPAD*, Denver, CO, NASA Conference Publication 2143, 1980, pp. 75-94.

26. Johnson, H.R., D.L. Comfort, and D.D. Shull, "An Engineering Data Management System for IPAD," *Proceedings of National Symposium for IPAD*, Denver, CO, NASA Conference Publication 2143, 1980, pp. 145-178.

27. Bernstein, T.N., "Integrated Design Support System," *Managing Engineering Data: The Competitive Edge*, The 1987 ASME International Computers in Engineering Conference and Exhibition, New York, NY, August 1987, pp. 41-47.

28. "Information Support System (IISS) Information Modeling Manual - Extended ($IDEF_{1X}$)," D. Appleton Company, Inc., Manhattan Beach, CA, 1985.

29. Gloudeman, J.F., and R.R. Brown, "Managing Engineering Analysis Data," *Computers in Mechanical Engineering*, March 1987, pp. 42-44.

30. Rouse, N.E., "Managing Engineering Databases," *Machine Design Magazine*, September 10, 1987, pp. 108-112.

31. Fenves, S.J., and W.J. Rasdorf, "Role of Data Management Systems in Structural Engineering," *Proceedings of the IABSE Colloquium on Informatics in Structural Engineering*, International Association of Bridge and Structural Engineers, October 1982, pp. 229-242.

32. Clemons, E.K., "Data Models and ANSI/SPARC Architecture," *Principles of Database Design*, Vol. 1, S. B. Yao, Ed., Prentice-Hall, Inc., 1985, pp. 66-114.

33. Benayoune, M., and P. E. Preece, "Methodology for the Design of Databases for Engineering Applications," *Journal of Computer-Aided Design*, Vol. 18, No. 5, June 1986, pp. 257-262.

34. Du, H.C., and S. Ghanta, "A Framework for Efficient IC/VLSI CAD Databases," *Proceedings of IEEE 1987 Third International Conference on Data Engineering*, pp. 619-625.

35. Cammarata, S.J., and M.A. Melkanoff, "An Information Dictionary for Managing CAD/CAM Databases," *Data Programming and Design*, Vol. 1, No. 3, March 1988.

36. Rice, L., "Managing CAD/CAM/CAE and Design Processes in Mechanical Engineering," *Proceedings in Computers in Mechanical Engineering*, pp. 421-428.

37. Mindlin, H., and S.H. Smith, "Database System Considerations in Engineering Design," *Managing Engineering Data: The Competitive Edge*, The 1987 ASME International Computers in Engineering Conference and Exhibition, New York, NY, August 1987, pp. 9-15.

38. Bray, O.H., "Engineering Database Management Requirements: The Next Generation," *Managing Engineering Data: The Competitive Edge*, The 1987 ASME International Computers in Engineering Conference and Exhibition, New York, NY, August 1987, pp. 41-47.

39. Rasdorf, W.J., "Extending DBMS's for Engineering Applications," *Computers in Mechanical Engineering*, March 1987, pp. 62-69.

40. Rasdorf, W.J., "Engineering Database Management Needs," *Management Engineering Data: The Competitive Edge*, The 1987 ASME International Computers in Engineering Conference and Exhibition, New York, NY, August 1987, pp. 41-47.

41. Ketabchi, M.A., and V. Berzins, "Modeling and Managing CAD Databases," *Computer*, February 1987, pp. 93-102.

42. ANSI/X3/SPARC, "Interim Report from the Study Group on Data Base Management Systems," 75-02-08, *FDT* (Bulletin of the ACM SIGMOD), Vol. 7, No. 2, 1975.

43. Hammer, M., and D. McLeod, "Database Description with SDM: A Semantic Database Model," *ACM Transactions on Database Systems*, Vol. 6, No. 3, September 1981, pp. 351-386.

44. Buneman, P., and R.E. Franke, "FQL--A Functional Query Language," *Proceedings of ACM SIGMOD International Conference on the Management of Data*, Boston, May 30-June 1, 1979.

45. Tsai, W., "Composite Design," Third Edition, Think Composite Company, Dayton, OH, 1985.

46. Mallick, P.K., "Fiber-Reinforced Composite," Marge Dekker Publishing Company, 1988.

47. Parks, C.H., "Tutorial: Reading and Reviewing the Common Schema for Electrical Design and Analysis," *Proceedings of 24th ACM/IEEE Design Automation Conference*, 1987, pp. 479-483.

48. Shunk, D., B. Sullivan, and J. Cahill, "Making the Most of IDEF Modeling - The Triple-Diagonal Concept," *CIM Review, The Journal of Computer-Integrated Manufacturing Management*, Vol. 3, No. 1, Fall 1986, pp. 12-17.

49. "Integrated Computer-Aided Manufacturing (ICAM) Architecture, Part II, Vol. IV - Functional Modeling Manual ($IDEF_0$)," Softech, Inc., Waltham, MA, 1981.

50. Mackulak, G.T., "High Level Planning and Control: an $IDEF_0$ Analysis for Airframe Manufacture," *Journal of Manufacturing Systems*, Vol. 3, No. 2, pp. 121-133.

51. Mackulak, G.T., "An Examination of the $IDEF_0$ Approach Used as a Potential Industrial Standard for Production Control System Design," *Automated*

Manufacturing, ASTM STP862, L. B. Gargner, Ed., Philadelphia, 1985, pp. 136-149.

52. "Integrated Computer-Aided Manufacturing (ICAM) Architecture, Part III, Vol. V - Information Modeling Manual ($IDEF_1$)," Softech, Inc., Waltham, MA, 1981.

53. Loomis, M.E.S., "Modeling - The $IDEF_{1X}$ Technique," *IEEE 1986 Phoenix Conference on Computers and Communications*, pp. 146-151.

54. Bravoco, R.R., and S.B. Yadav, "A Methodology to Model the Information Structure of an Organization," *The Journal of Systems and Software*, Vol. 5, 1985, pp. 59-71.

55. Bravoco, R.R., and S.B. Yadav, "A Methodology to Model the Dynamic Structure of an Organization," *Information Systems*, Vol. 10, No. 3, 1985, pp. 299-317.

56. "Integrated Computer-Aided Manufacturing (ICAM) Architecture, Part III, Vol. VI - Dynamics Modeling Manual ($IDEF_2$)," Softech, Inc., Waltham, MA, 1981.

57. Southall, J.W., "Development of Integrated Programs for Aerospace-Vehicle Design (IPAD) - Information Processing Requirements," NASA Contract NAS1-14700, NASA Contract Report CR-2981, Boeing Commercial Airplane Company, June 1977.

58. Nijssen, G.M., "Databases of the Future: Design and Methodology," *Database Journal (GB)*, Vol. 12, No. 1, January 1982, pp. 2-9.

59. Flynn, D.J., and A.H.F. Laender, "Mapping from a Conceptual Schema to a Target Internal Schema," *The Computer Journal (UK)*, Vol. 28, No. 5, 1985, pp. 508-516.

60. Gadre, S.H., "Building an Enterprise and Information Model," *Database Programming and Design*, Premier Issue 1987, San Francisco, CA, 1987, pp. 48-58.

61. Shoval, P., "Essential Information Structure Diagrams and Database Schema Design," *Information Systems*, Vol. 10, No. 4, 1985, pp. 417-423.

62. Chen, P.S., "Database Design Based on Entity and Relationship," *Principles of Database Design*, Vol. I, edited by S. B. Yao, 1985, pp. 174-210.

63. Chen, P.S., "The Entity-Relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976, pp. 9-36.

64. Atwood, T.M., "An Object-Oriented DBMS for Design Support Applications," *Proceedings of IEEE Compint - Computer Aided Technologies*, 1985, pp. 299-307.

65. Dittrich, K.R., and R.A. Lorie, "Object-Oriented Database Concepts for Engineering Applications," *Proceedings of IEEE Compint - Computer Aided Technologies*, 1985, pp. 321-325.

66. Landis, G.S., "Design Evolution and History in an Object-Oriented CAD/CAM Database," *IEEE Computer Society Compcon, Spring '86*, March 1986, pp. 297-303.

67. Spooner, D.L., Michael and D.B. Faatz, "Modeling Mechanical CAD Data Abstraction and Object-Oriented Techniques," *Proceedings: IEEE 1986 International Conference on Data Engineering*, 1986, pp. 416-424.

68. McLeod, D., K. Narayanaswamy, and K.V. Bapa Rao, "An Approach to Information Management of CAD/VLSI Applications," *Proceedings of 1983 ACM SIGMOD/IEEE: Engineering Design Applications*, 1983, pp. 39-49.

69. Stonebraker, M., B. Rubenstein, and A. Guttman, "Application of Abstract Data Types and Abstract Indices to CAD Databases," *Proceedings of Annual Meeting: Database Week*, IEEE Computer Soc. Press, 1983, pp. 107-113.

70. Derret, N.P., D.H. Fishman, W. Kent, P. Lyngback, and T.A. Ryan, "An Object-Oriented Approach to Data Management," *IEEE Computer Society Compcon, Spring '86*, March 1986, pp. 330-335.

71. Pascoe, G.A., "Elements of Object-Oriented Programming," *Byte*, August 1986, pp. 139-144.

72. Fenves, G.L., "Object Representations for Structural Analysis and Design," *Proceedings of Computers in Civil Engineering*, 1988, pp. 502-511.

73. LeJacq, J.P., "Functional Requirements for CAD Database Using a General Modeling Abstraction of Physical Systems," *Conference on Data and Knowledge Systems for Manufacturing and Engineering*, October 1987, pp. 111-114.

DISTRIBUTION
IDA PAPER P-2154

MANAGING ENGINEERING DESIGN INFORMATION

133 Copies

	Number of Copies
Department of Defense	
OUSD (R&AT)/ET Rm. 3D1089, Pentagon Washington, DC 20301-3080	
ATTN: Mr. Raymond Siewert	1
Dr. Leo Young	1
Assistant Deputy, Systems, OASD/P&L Department of Defense Room 2B322, Pentagon Washington, DC 20301-8000	
ATTN: Mr. Russell R. Shorey	1
OASD CP&LS WS Rm. 2B322, The Pentagon Washington, DC 20301-8000	
ATTN: Mr. Don Hall	1
Office of the Secretary of Defense (P&L) Director, CALS Rm. 2B322, Pentagon Washington, DC 20301	
ATTN: Michael McGrath	1
OASD (P&L), Systems Director, Weapon Support Improvement Group Rm. 2B322, Pentagon Washington, DC 20301	
ATTN: Martin Meth	1
Dr. William E. Isler Program Manager, ISTO Defense Advanced Research Projects Agency 1400 Wilson Boulevard, 7th Floor Arlington, VA 22209-2308	1

Dr. Jacob Schwartz Director, Information Science and Technology Office Defense Advanced Research Projects Agency 1400 Wilson Boulevard, 7th Floor Arlington, VA 22209-2308	1
Dr. Stephen L. Squires Information Science and Technology Office Defense Advanced Research Projects Agency 1400 Wilson Boulevard, 7th Floor Arlington, VA 22209-2308	1
Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2

Miscellaneous, U.S. Government

Dr. Jaroslaw Sobieski Deputy Head, Interdisciplinary Research Office NASA Langley Research Center Structures Directorate Hampton, VA 23665	1
Dr. John Zuk 1 Chief, Advanced Plans and Programs Office NASA Ames Research Center Moffett Field, CA 94035	
National Science Foundation Directorate for Engineering Division of Design and Manufacturing Systems 1800 G Street, NW Washington, DC 20550	
ATTN: Dr. Marvin DeVries	1
Dr. Branimir F. von Turkovich	1

Department of the Army

Mr. Sid Markowitz US Army AMCCOM ATTN: AMSMC-QAHP Picatinny Arsenal, Bldg. 62 Dover, NJ 07806-5000	1

Mr. Geza Papp 1
Chief of Technology
US Army AMCCOM
Picatinny Arsenal, Bldg. 62
Dover, NJ 07806-5000

Mr. Charles Reading 1
Chief, Logistics Management Division
Cdr USAAVSCOM
AMCPM-LHX-L
4300 Goodfellow Boulevard
St. Louis, MO 63120-1798

<u>Department of the Navy</u>

Mr. W. J. Willoughby 1
Director for Reliability, Maintainability
and Quality
Office of the Assistant Secretary of the Navy
(Shipbuilding and Logistics)
Washington, DC 20360-5000

Department of the Navy 1
Assistant Secretary of the Navy
Research, Engineering, and Systems
Rm. 4E732, Pentagon
Washington, DC 20350-1000

Mr. Albert Knight 1
Naval Ocean Systems Center
Computer Integrated Engineering, Code 936
San Diego, CA 92152-5000

<u>Department of the Air Force</u>

Hon. John J. Welch 1
Assistant Secretary of the Air Force (Acquisition)
HQ, US Air Force
Rm. 4E964, Pentagon
Washington, DC 20330-1000

Brig. Gen. W. E. Collins 1
Special Assistant for Reliability and Maintainability
HQ, USAF/LE-RD
Rm. 4E259, Pentagon
Washington, DC 20330

Col. James Harrington HQ USAF/LE-RD Rm. 4E259, Pentagon Washington, DC 20330			1
Dr. John Dimmock Technical Director Air Force Office of Scientific Research Bolling AFB, Bldg. 410 Washington, DC 20332			1
Capt. Maureen Harrington AFC SA/SAGR Washington, DC 20330-5420			1
AFHRL/LRL Wright-Patterson AFB, OH 45433			9
ATTN:	Capt. Raymond Hill	5	
	Capt. Mike Hanuschik	1	
	Capt. Rick Berry	1	
	Capt. Don Loose	1	
	Ms. Wendy Campbell	1	
Dr. Gary Denman Deputy Director Wright Research and Development Center Area B, Bldg. 45, Rm. 207 Wright-Patterson AFB, OH 45433			1
Mr. Alan E. Herner Operations Research Analyst WRDC/MTC Wright-Patterson AFB, OH 45433			1
Mr. Raymond L. Johnson Technical Director Aeronautical Systems Division USAF, ASD/ENS Directorate of Systems Engineering Wright-Patterson AFB, OH 45433-6503			1
Dr. Squire Brown Director, Technology Assessment Office AFWAL/FIA Wright-Patterson AFB, OH 45433			1

Academia

Dr. Alice M. Agogino | 1
Associate Professor of Mechanical Engineering
Mechanical Engineering Department
University of California at Berkeley
5136 Etcheverry Hall
Berkeley, CA 94720

Dr. Erik K. Antonnson | 1
Assistant Professor of Mechanical Engineering
Engineering Design Research Laboratory
California Institute of Technoogy
Mail Code 104-44
Pasadena, CA 91125

Dr. Richard J. Balling | 1
Civil Engineering Department, 368 CB
Brigham Young University
Provo, Utah 84602

Dr. Ben Blanchard | 1
Assistant Dean of Engineering for Extension
Virginia Polytechnic and State University
College of Engineering
341 Norris Hall
Blacksburg, VA 24061

Dr. Dale E. Calkins | 1
Research Associate Professor
Ocean Engineering Program
Department of Mechanical Engineering
University of Washington
Seattle, WA 98195

Dr. Alexander N. Christakis | 1
Defense Systems Management College
Fort Belvoir, VA 22060-5426

Dr. Don Clausing | 1
Laboratory for Manufacturing and Productivity
Massachusetts Institute of Technology
Bldg. 35, Room 238
77 Massachusetts Avenue
Cambridge, MA 02139

Dr. John R. Dixon 1
Professor of Mechanical Engineering
University of Massachusetts
Department of Mechanical Engineering
Engineering Laboratory Building
Amherst, MA 01003

Dr. Walter J. Fabrycky 1
Professor of Industrial Engineering and
Operations Research
Virginia Polytechnic and State University
Blacksburg, VA 24061

Dr. Susan Finger 1
Carnegie-Mellon University
The Robotics Institute
Pittsburgh, PA 15213

Dr. Mark Fox 1
Director, Intelligent Systems Laboratory
Carnegie-Mellon University
The Robotics Institute
Pittsburgh, PA 15213

Dr. Robert E. Fulton 1
Professor, Department of ME
Georgia Institute of Technology
Mechanical Engineering Department
Atlanta, GA 30332

Dr. M. Hardwick 1
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Dr. Michel A. Melkanoff 1
University of California at Los Angeles
3532 Boelter Hall
Los Angeles, CA 90024-1597

Dr. Farrokh Mistree 1
Professor of Mechanical Engineering
Mechanical Engineering Department
University of Houston
Houston, TX 77004

Mr. Thomas J. Murrin 1
Distinguished Service Professor of Technology
Carnegie Mellon University
4630 Wean Hall
Pittsburgh, PA 15213-3890

Dr. Gerald Nadler 1
Chairman, Department of Industrial and Systems
Engineering
Industrial and Systems Engineering
University of Southern California
University Park
Los Angeles, CA 99089-1452

Dr. Michael Pecht 1
Associate Professor of Mechanical Engineering
Mechanical Engineering Department
University of Maryland
College Park, MD 20742

Dr. Robin J. Popplestone 1
Director, Laboratory for Perceptual Robotics
Department of Computer and Information Science
University of Massachusetts
Lederly Graduate Research Center
Amherst, MA 01003

Dr. Ramana Reddy 1
Professor of Computer Science, Dir., AI Laboratory
Department of Statistics and Computer Science
West Virginia University
Knapp Hall
Morgantown, WV 26506

Mr. Edward Rogan 1
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150

Dr. Bela I. Sandor 1
Professor of Engineering Mechanics
University of Wisconsin - Madison
2359 Engineering Building
Madison, WI 53706

Dr. Daniel Schrage 1
Professor, School of Aerospace Engineering
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Dr. D. L. Spooner 1
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Dr. Stanley Y.W. Su 1
Data Base Systems Research and
Development Center, 470 CSE
University of Florida
Gainesville, FL 32611

Dr. Edison T.S. Tse 1
Director, Decision Systems Laboratory
Department of Engineering-Economic Systems
Stanford University
Stanford, CT 94305

Dr. Manjula B. Waldron 1
Professor, Department of Biomedical Engineering
The Ohio State University
2070 Neil Avenue
Columbus, OH 43210

Dr. John N. Warfield 1
Director, Institute for Advanced Study in the Integrative
George Mason University
4400 University Drive, 219 Thompson Hall
Fairfax, VA 22030-4444

Dr. Raymond Willem 1
Associate Professor, Mechanical Engineering
New Mexico State University
Box 345D
Las Cruces, New Mexico 88003

Dr. Michael J. Wozny 1
Rensselaer Design Research Center
Rensselaer Polytechnic Institute
CII, Rm. 7015
Troy, NY 12180-3590

Dr. Chou-pin Yeh 1
Mechanical Engineering Department
Georgia Institute of Technology
Atlanta, GA 30332

<u>Industrial Organizations</u>

Mr. Michael G. Adami 1
Chief, CAD/CAM Methodology
Research and Enginering Department
United Technologies Sikorsky Aircraft
6900 Main Street, MS S308A
Stratford, CT 06601-1381

Mr. William G. Beazley 1
Editor and Publisher, CALS Report
Knowledge Base, International
13939 Northwest Freeway, Suite 270
Houston, TX 77040

Mr. George Beiser 1
3001 N. Florida Street
Arlington, VA 22207

Mr. Timothy V. Bogard 1
Manager, Design Automation and Engineering Services
Texas Instruments
8505 Forrest Lane, P.O. Box 660246
Dallas, TX 75266

Ms. ML Brei 1
8509 Rayburn Road
Bethesda, MD 20817

Mr. E. Perley Eaton 1
Principal Engineer, Federal Systems
TRW Defense Systems
Command Support Division
1 Federal Systems Park
12900 Fair Lakes Parkway
Fairfax, VA 22033

Mr. Mark Erdrich 1
Laboratory Head
Adv. CAD/CAM/CAE
Eastman Kodak Company
7/23 Kodak Park
Rochester, NY 14650

Dr. Iman Foroutan 1
Chief, Computer Integrated Manufacturing
Microelectronic Circuits Division
Industrial Electronics Group
Hughes Aircraft Company
P. O. Box H, 500 Superior Avenue
Newport Beach, CA 92658-8903

Mr. David A. Gatenby 1
Supervisor, CAE/D and DFX Planning
AT&T
One Oak Way, Rm. 3EC126
Berkeley Heights, NJ 07922

Mr. Siegfried Goldstein 1
Siegfried Enterprises, Inc.
P. O. Box 2308
North Babylon, NY 11703

Ms. Maria W. Huang 1
Data Asset Manager, Information Resources
FMC Corporation
Ground Systems Division
881 Martin Avenue, Box 58123
Santa Clara, CA 95052

Dr. Reid June 1
Manager, Structures Technology
Boeing Commercial Airplane Company
P.O. Box 3707, MS 4A-06
Seattle, WA 98124

Dr. Clinton Kelly 1
Corporate Vice President for Advanced
Technology Programs
SAIC, Inc.
1710 Goodridge Drive, Rm. 1408
McLean, VA 22102

Mr. Anthony Kim 1
Manager, Aircraft Performance Methods
General Electric
Aircraft Engine
One Neumann Way
Cincinnati, OH 45215-6301

Mr. C. T. Kitzmiller 1
Advanced Technology Center
Boeing Computer Services
P.O. Box 24346, MS 7L-64
Seattle, WA 98124-0346

Mrs. Mary Klement 1
Engineering Specialist
General Dynamics
P.O. Box 85357, MS P1-2610
San Diego, CA 92138

Dr. Janusz Kowalik 1
Engineering Technology Applications
Boeing Computer Services
P.O. Box 24346
Belleview, WA 98124-0346

Dr. Manuel Lores 1
Director, Aircraft Technology
Lockheed Aeronautical Systems Company
Department 01-35, Bldg. 63-2
Plant A-1, P.O. Box 551
Burbank, CA 91520-0135

Mr. Stephen A. Magnus 1
Editorial Assistant, CAD/CAM Alert
Management Roundtable, Inc.
824 Boylston Street
Chestnut Hill, MA 02167

Mr. Stephen Meyer 1
McDonnell Douglas Helicopter Company
5000 E. McDowell Rd., Bldg. 530/B220
Mesa, AZ 82505-9797

Dr. Alan Mitchell 1
Director, Preliminary Design Tool Development
Research and Engineering Division
Boeing Aerospace Company
P. O. Box 3999, MS 82-23
Seattle, WA 98124-5214

Mr. James L. Nevins 1
Division Leader
The Charles Stark Draper Laboratory, Inc.
Robotics and Assembly Systems Division
555 Technology Square
Cambridge, MA 02139

Mr. Guy B. Olney 1
Boeing Computer Services
P.O. Box 24346, M/S 7L-62
Seattle, WA 98124

Mr. David Owen 1
NTL, Inc.
P. O. Box 597
Northport, NY 11768

Mr. Joseph Piteo 1
United Technologies
Sikorsky Aircraft
MS S308A
6900 Main Street
Stratford, CT 06601-1381

Mr. Charles R. Runyan Assistant Manager, Engineering Division Hughes Aircraft Company Radar Systems Group P. O. Box 92426, Bldg. R1 Los Angeles, CA 90009	1
Dr. Kamar J. Singh Project Manager, DARPA DICE General Electric Company One Neumann Way, MD H85 Cincinnati, OH 45215-6301	1
Mr. David F. Zarnow Assistant Manager, Hybrid Engineering Dept. Hughes Aircraft Co., Solid State Products Div. Microelectronic Circuits Division, MS A2202 500 Superior Avenue, Bldg. 700 Newport Beach, CA 92658-8903	1
Institute for Defense Analyses 1801 N. Beauregard Street Alexandria, VA 22311	35

Gen. William Y. Smith	1
Mr. Philip L. Major	1
Dr. Robert E. Roberts	1
Dr. William J. Schultis	1
Dr. Robert I. Winner	1
Mr. William E. Cralley	5
Mr. David A. Dierolf	1
Mr. G. Watts Hill	1
Dr. James Pennell	1
Dr. Karen J. Richter	10
Dr. Frederick R. Riddell	1
Dr. Edwin S. Townsley	1
Control and Distribution	10